Radio

elettronica & Computer

Anno XVI - Numero 5 - Sped. in abb. post. Gr. III/70%

MAGGIO/GIUGNO 1987 - L. 7.000

9 programmi su cassetta per C64 e C128

- FAI DA TE/L'HARDWARE PER CHI FOTOGRAFA
- GRAFICA/MERAVIGLIE CON L'HI-RES
- ANATOMIA/VIAGGIO ALLUCINANTE
- HOPELESS/SOLI NELLO SPAZIO!
- INGLESE/ULTIMA PUNTATA
- MULTIRAM/PROGRAMMI ATTIVI NELLO STESSO ISTANTE
- SOFTWARE/VENTICINQUE ASSALTI AL MURO MUTANTE
- GIOCO/FUGHE E CAPRIOLE INSEGUITI DA UN ROBOT
- SCRITTE/IN MOVIMENTO E GRANDI QUANTO VUOI

TASSA PAGATA PER CAMPIONE ALLEGATO

CON UN GRAFICO È MEGLIO
Super programma di business graphics

DIREZIONE GENERALE E AMMINISTRAZIONE

Editronica srl

20122 Milano-C.so Monforte 39
Ufficio abbonamenti 02/702429

Radio Elettronica & Computer

Direttore Responsabile
Stefano Benvenuti

Coordinamento editoriale
Francesca Marzotto

Impaginazione elettronica
Adelio Barcella

Collaboratori
Eleonora Boffelli
Giorgio Caironi
Mirko Diani
Marco Gussoni
Mario Magnani
Dolma Poli
Ivonne Rossi

SERVIZIO ABBONAMENTI
Editronica srl - C.so Monforte 39 - Milano Conto Corrente Postale n. 19740208 Una copia L. 6.000 - Arretrati: il doppio del prezzo di copertina. Abbonamento 12 numeri L. 60.000 (estero L. 80.000). Periodico mensile. Stampa: "VEGA sas" Via Teod[illegible]io 17, Milano. Distribuzione esclusiva per l'Italia A.&C. Marco S.p.A. - Via Fortezza 27 - 20126 Milano. Tel. 02/25261 - Telex 350320. Registrazione Tribunale di Milano N. 112/72 del 17.3.72. Pubblicità inferiore al 70%

RadioELETTRONICA & COMPUTER è titolare in esclusiva per l'Italia dei testi e dei progetti di Radio Plans e Electronique Pratique, periodici del gruppo Société Parisienne d'Edition.

SOMMARIO

MAGGIO-GIUGNO 1987 - ANNO XVI - NUMERO 5

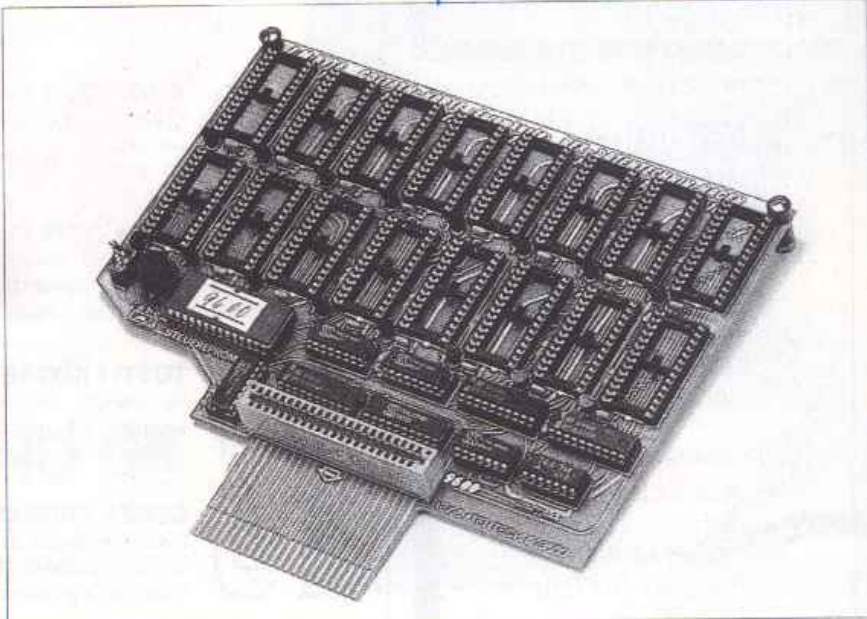
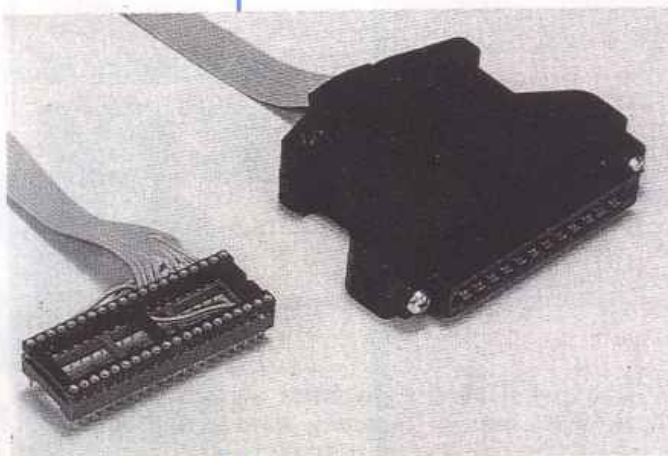
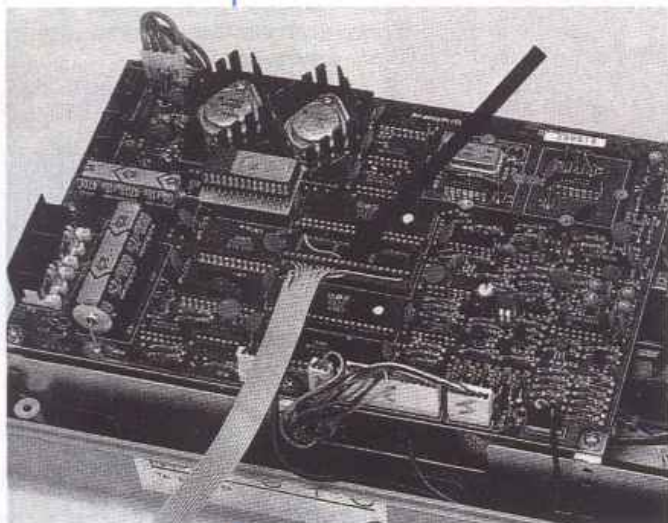
Le istruzioni per il caricamento della cassetta allegata a questo numero di *RadioElettronica & Computer* sono a pag. 50.

Chi, Cosa, Come, Quando...

Backup veloce

Esistono vari programmi di duplicazione per il C64. Spesso però sono lenti, non girano se ci sono blocchi difettosi o lavorano solo con un hardware acceleratore. Il sistema Copy+ risolve tutti questi inconvenienti, poiché duplica anche i settori difettosi, ed è facilissimo da realizzare; basta un intervento sulla unità disco: bisogna sollevare dal suo zoccolo la CIA 6522 di destra (**figura 1**) per incorporare il cavo parallelo; si inserisce quindi lo zoccolo di adattamento con il cavo brasato, e si rimette l'IC al suo posto. Ecco l'occorrente: 30 cm di cavo piatto a 10 vene, uno spinotto da porta utente, 2 zoccoli IC a 40 poli. Il materiale (eventualmente anche il cavo già brasato) si trova nei negozi di accessori per computer con modica spesa. Il collegamento da fare con il cavo fra la CIA 6522 e la porta utente (per esempio con la spina) è mostrato in **figura 2**.

Figura 1: come si incorpora il cavo parallelo.

Figura 2 (in basso): collegamento con la porta utente.

Il gigante della memoria

Con una capacità di memoria di 1 Megabyte, la piastra Goliath apre una nuova dimensione per le cartridge di Eprom: consente di tenere in linea sulla porta di espansione il contenuto di ben cinque dischetti per una consultazione quasi istantanea.

Poiché il programma di controllo della piastra viene scaricato da $C000 in avanti, tutti i programmi che risiedono in quest'area devono essere rilocati altrove.

Per i programmatori di linguaggio macchina esiste uno schema per le rilocazioni tipiche, illustrato nel libretto istruzioni (i comandi JMP non vengono trasformati).

Con il gestore del modulo integrato nel software di controllo è possibile realizzare dei file per i diversi tipi di Eprom, pronti per essere riempiti e scaricati in memoria o su un dischetto. Inoltre è molto semplice mettere in lista e far partire tutti i programmi composti dal gestore dei moduli, e che in seguito vengono inseriti in una Eprom.

E' possibile sezionare le Eprom una per una in blocchi da 8 Kb, quindi possono essere utilizzate anche Eprom non generate dal gestore del Goliath.

Il costo della piastra, non ancora distribuita in Italia, è di 149 marchi tedeschi. Per ulteriori informazioni rivolgersi a Rex Datentechnik, Stresemannstr. 11, 58 Hagen 1, Tel. 02331 32734 Germania Ovest.

Concorso software

Il circolo culturale "P. Giuseppe Airoldi", in collaborazione con il Comune di Lissone e Computerlandia, organizza il 1° Concorso a premi per software hobbistico su C64, aperto a tutti gli appassionati creatori di programmi originali e articolato in tre sezioni (didattica, giochi e generale).

I concorrenti possono partecipare con un programma di loro creazione per ogni sezione.

Sarà premiato il miglior programma di ciascun settore con un modem per C64 (valore 100 mila lire) fornito di relativo software e saranno assegnati 4 premi speciali a programmi di particolare originalità.

A tutti i concorrenti verrà assegnato un premio di partecipazione.

I tagliandi di iscrizione, che possono essere richiesti insieme al regolamento al Circolo culturale Airoldi, Concorso Software, casella postale 119, 20035 Lissone (MI) oppure telefonicamente al numero 039 795474, dovranno pervenire entro il 31 maggio 1987 insieme al programma con relativa documentazione e alla quota di iscrizione.

Le premiazioni avranno luogo il 14 giugno 1987 presso la sede del circolo.

Commodisk 11

Commodisk n. 11 è in edicola e come sempre offre cinque programmi su dischetto a sole 13 mila lire.

In questo numero si completa il Sistema Autore, con le parti relative alla costruzione dei corsi e alla loro interpretazione. Sono fornite le indicazioni per realizzare un dischetto mas ter comprendente tutte le facilities di questo pacchetto.

Il gestionale del mese è un potentissimo programma per la gestione della contabilità di magazzino, utilizzabile a se stante oppure interfacciato al programma di contabilità pubblicato sul n. 6. Anche qui ci sono le indicazioni per costituire il dischetto con il pacchetto integrato.

Il gioco, spettacolare come sempre, è ambientato sulle spiagge di un'isola dei Caraibi, dove una guarnigione implacabile difende strenuamente il proprio quartier generale.

Il quarto programma consente di progettare e stampare poster di dimensione variabile e con una risoluzione grafica accuratissima. Infine l'utility, Quick Dos, mette a disposizione, in 49 blocchi di puro linguaggio macchina, tre favolosi tool integrati: un confortevole ambiente di sviluppo per il Basic, un potentissimo monitor per il linguaggio macchina e un comodo text editor. Il tutto senza togliere un solo byte all'area Basic.

Colori da Las Vegas

Anche l'Atari è uscita a Las Vegas con il suo Ibm compatibile, il Mega ST.

Senza rinunciare alle sue prerogative grafiche e di interfaccia user friendly l'Atari propone un classico compatibile MS-DOS completo però di molti optional di grande comfort.

Caratteristiche: 512K RAM espandibile a 640 su scheda madre, 256K dedicati per la gestione del video, scheda grafica personalizzata comprendente le famose EGA, CGA ed Hercules, un disk drive interno da 5" e 1/4, interfacce parallela e seriale 232 standard.

Vengono offerti anche un mouse e il sistema operativo GEM. Possibilità di espansione praticamente illimitate: drive esterni da 5" e 1/4, da 3" e 1/2, hard disk di varia capacità, modem, eccetera.

La tastiera è quella tipica a 84 tasti ed è identica a quella dell'IBM XT.

Il nuovo Atari sarà disponibile in Italia per la fine di luglio a circa 1.600.000 lire (Iva esclusa) nella versione con monitor a colori e a circa 1.200.000 (Iva esclusa) con monitor monocromatico.

Per ulteriori informazioni è possibile rivolgersi direttamente alla Atari Italia, Via dei Lavoratori 19, Cinisello (Milano), telefono 02 6120851/4.

Logo per il C64
D. Watt, editore McGrawHill, 32 mila lire.

Il Logo, linguaggio sviluppato al MIT e destinato alla prima alfabetizzazione informatica, si è imposto non solo come strumento per stimolare i bambini a familiarizzare con il computer ma anche come ambiente educativo globale. Il libro introduce le semplicissime regole di questo linguaggio attraverso numerose attività, facilmente realizzabili anche dai più giovani lettori, da soli o con l'aiuto dei genitori, applicate al Commodore 64, attualmente il più diffuso degli home computer. Il Logo per il Commodore 64 può essere utilizzato come manuale per l'insegnamento familiare o scolastico e contiene, suddivisa nei vari capitoli, una guida preziosa per chi accompagna i bambini nella loro scoperta del computer e del mondo della logica. Ogni aspetto che rappresenta una potenziale difficoltà viene evidenziato per aiutare i lettori a muoversi in questo linguaggio: l'impostazione volutamente elementare dei comandi Logo, infatti, non deve trarre in inganno, perché questo approccio estremamente amichevole nasconde la possibilità di sviluppare programmi che non hanno nulla da invidiare a quelli realizzati con linguaggi ritenuti più potenti.

A rette, a barre o a torta?

Un programma gestionale che si rispetti deve consentire una visione immediata e di insieme della situazione. Quale strumento è più efficace di un generatore di grafici orientati al business? Questo programma consente la rappresentazione grafica di una serie qualunque di valori.

In una biblioteca di programmi che si rispetti non ne può certamente mancare uno che dia la possibilità di disegnare grafici matematici, il cui andamento dipende dall'inserimento di valori numerici arbitrari.

Questo programma, infatti, vi permette di visualizzare facilmente la rappresentazione grafica di una qualunque serie di valori: dopo aver inserito i dati che vi servono per un determinato calcolo, comparirà immediatamente il grafico relativo all'andamento di tali valori, disponibile nelle tre forme dell'istogramma, della torta percentuale e delle rette.

Oltre all'inserimento di un qualunque dato numerico arbitrario, SuperGraf vi offre anche la possibilità di utilizzare, nella rappresentazione grafica che compare dopo la fase di calcolo, i valori con i quali avete lavorato su altri programmi, caricando una figura o un file.

Per quanto riguarda la fase di stampa, poi, SuperGraf presenta parecchie opzioni al riguardo, che vi permettono di stampare i grafici relativi ai dati che avete precedentemente inserito nel computer.

Come funziona il programma

Dopo aver caricato il programma SuperGraf, comparirà uno schermo raffigurante 3 colonne, ognuna costituita da 20 righe (quantità modificabile tramite gli appositi comandi), all'interno delle quali potrete inserire il valore, pressoché illimitato, dei dati dei quali volete osservare la rappresentazione grafica.

In pratica, potrete inserire nella parte inferiore dello schermo, a fianco del cursore lampeggiante, il valore di 60 dati, suddivisi in tre grafici, in modo tale che ogni rappresentazione grafica presenti 20 valori numerici.

Durante questa fondamentale fase di programmazione e di predisposizione del programma a rappresentare graficamente i dati da voi richiesti, potrete consultare un'apposita schermata di aiuto, richiamabile tramite il tasto F3, che vi elencherà tutti i comandi disponibili e le operazioni ad essi relative.

Poiché questo schermo comprende tutti i comandi del SuperGraf, va da sé che è più semplice e comodo, ai fini di una sua dettagliata descrizione, riferirsi a tale schermata di aiuto, procedendo nello stesso ordine seguito dai programmatori del Commodore che hanno ideato questo programma.

Per quanto riguarda la fase di inserimento dei dati, potrete utilizzare i tasti CRSR per muovere il cursore secondo le solite modalità, F7 e F8 per spostarvi rispettivamente da una colonna all'altra verso destra e sinistra, INST/DEL per inserire o eliminare dei caratteri vicini al cursore, CLR/HOME e SHIFT+CLR/HOME per portare immediatamente il cursore alla prima riga in alto a sinistra e all'ultima in basso a destra e Logo Q (tasto Commodore + Q) per passare alla schermata grafica.

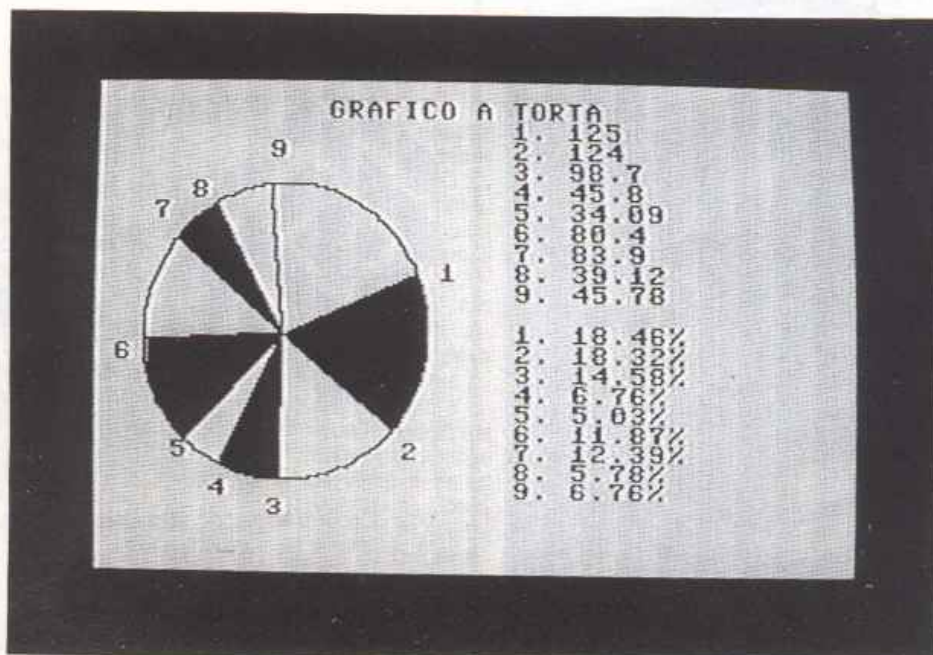
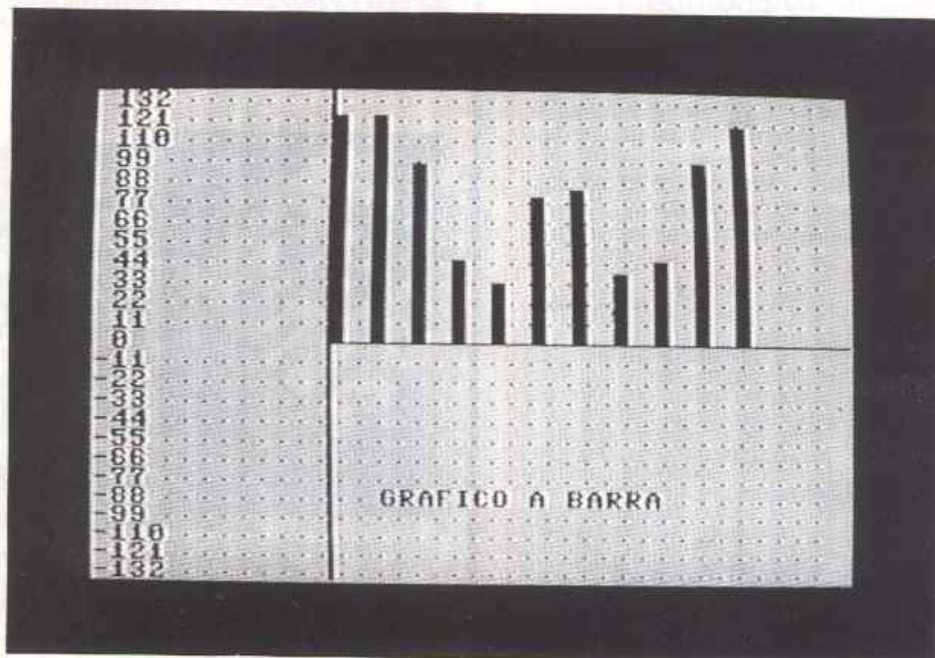
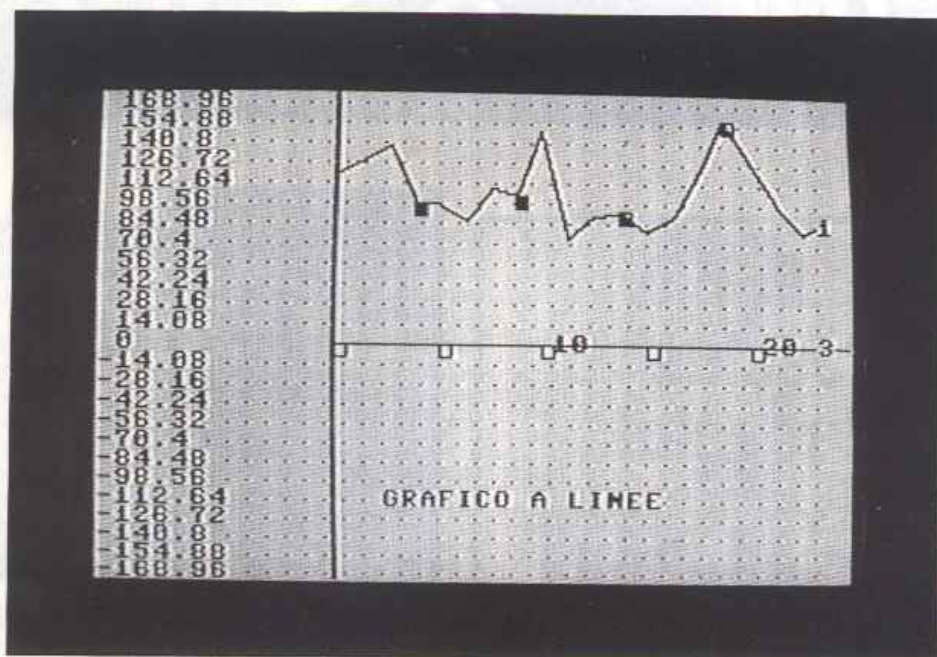
Inoltre, sempre nello schermo relativo all'inserimento dei dati, ma nel modo Comandi computer, contrassegnato da una C vicino al cursore (invece che da una N, come per i comandi cui si è accennatoprecedentemente),potreteregistrare un grafico (SG'nome') o un file (SD'nome'), caricare un grafico (CG'nome') o un file (CD'nome'), cancellare un file (CF'nome'), caricare il programma in riferimento a determinate righe o colonne (L3'nome'; riga'numero'; colonna'numero'), o ai caratteri ASCII (LA'nome'), e, infine, predisporre il programma per l'uso delle stampanti CBM 1525 e MPS 801 (STAMP 1;) e CBM 1526 e MPS 802 (STAMP 2;).

Un altro aiuto

Premendo il tasto Return (Q + RETURN vi fa tornare al Modo comandi), poi, potrete passare alla seconda schermata di aiuto, che vi fornirà ulteriori indicazioni sui comandi utilizzabili nel modo C, del quale si è già accennata la funzione.

Digitando COPI R; C; potrete copiare il dato presente in riga R e colonna C nella posizione in cui si trova l'icona luminosa, e digitando ROTARE potrete ruotare le colonne (cambiandone la disposizione verso destra).

Potrete inoltre formattare un nuovo dischetto (con ID), richiamare la directory del dischetto inserito nell'unità-

drive (con DI), cancellare i dati inseriti nella memoria del programma (con NUOVO), passare allo schermo relativoallarappresentazionegraficadell'andamento dei dati inseriti nel computer (con BITS), richiamare i 3 diversi tipi di grafici (istogrammi con BAR, torte percentuali con TOR e rappresentazione tramite rette e segmenti con GRAFO), cancellare i grafici ottenuti (con ZERO), ripartire dall'inizio del programma (con RESET)e, infine, stampare le matrici dei dati (LISTA e con *P).

Rappresentazione grafica

Dopo aver inserito i valori che desiderate all'interno della prima schermata, di fianco al cursore lampeggiante nel modo N (ricordatevi che per passare da un modo all'altro è sufficiente premere il tasto F1), potrete immediatamente osservare la loro rappresentazione grafica scrivendo BITS dopo aver premuto il tasto F1 per passare al modo C (Comandi).

Inoltre, una volta passati allo schermo grafico, potrete richiamare i 3 diversi tipi di grafici (gli istogrammi premendo il tasto B, la torta percentuale premendo P e le rette con il tasto G), se già non lo avete fatto nel modo C con i relativi comandi.

Per quel che riguarda i comandi del Modo grafica, potrete consultare la terza e ultima schermata di aiuto, all'interno della quale, oltre ai diversi tipi di grafico, avrete modo di spostare il cursore verso destra (premendo il tasto R), verso sinistra (premendo L), in alto (U) e in basso (D).

Questi movimenti vi saranno particolarmente utili per definire un punto particolare all'interno del diagramma visualizzato sullo schermo, utilizzabile per i comandi ai quali si accennerà più avanti.

Infatti, utilizzando il comando Logo T (tasto Commodore + T), potrete inserire sullo schermo i caratteri che desiderate, premendo Logo 1 (tasto Commodore + 1) avrete modo di determinare un punto all'interno dello schermo, mentre con Logo 2 (tasto Commodore + 2) e con Logo C (tasto Commodore + C) avrete modo di tracciare, nell'ordine, una retta o un cerchio, il cui punto di partenza (per la prima) e il cui centro (per il secondo) sono individuati dal punto precedentemente determinato con il comando Logo 1 .

Infine, utilizzando l'opzione Logo P avrete modo di stampare direttamente il grafico che avete ottenuto, mentre con Logo F (tasto Commodore + F) potrete riempire, colorandola, la zona individuata in quel momento dal

cursore (ad esempio, una sezione del grafico torta percentuale, per contraddistinguerla dalle altre).

Scelta del grafico

Per quanto riguarda la fase di rappresentazione grafica delle serie di dati inseriti nel computer, vi conviene richiamare un grafico alla volta, ossia osservare dapprima, ad esempio, l'andamento delle rette relativo ai vostri dati, quindi cancellare con gli appositi comandi (consultate, se avete dubbi in proposito, la schermata di aiuto) il grafico così ottenuto e passare alla rappresentazione per istogrammi, e solo in seguito, effettuando una seconda volta le operazioni di cancellamento del grafico, osservare la torta percentuale.

Questo procedimento potrebbe apparire un po' laborioso; in realtà bisogna considerare il fatto che i grafici vengono visualizzati sul monitor sovrapposti fra di loro (nella parte destra gli istogrammi e le rette, mentre in quella sinistra la torta percentuale).

Pertanto, nel caso in cui abbiate lavorato su una notevole quantità di dati, vi risulterà problematico osservare attentamente la loro rappresentazione grafica nei minimi particolari.

Nella parte sinistra dello schermo del modo grafica, inoltre, potrete consultare in ogni momento i valori da voi introdotti nella memoria del programma, con a fianco il rapporto percentuale ad essi riferito rispetto al totale (se avete richiamato il grafico a torta tali valori compariranno nella parte destra dello schermo).

Infine, va segnalato che può accadere di dover attendere pazientemente qualche secondo prima di ogni rappresentazione grafica, dato che a volte sembra che il computer non abbia accettato i comandi: in realtà questi momenti sono interamente occupati dall'elaborazione dei dati che effettua direttamente SuperGraf.

Senza dubbio la facilità d'uso di questo programma, la sua estrema completezza di comandi e la vasta gamma di impieghi cui si presta vi permetteranno di utilizzare in ogni occasione (anche la più impensabile) il SuperGraf: d'altra parte, se casualmente non doveste ricordarvi alcune procedure o non riusciste a proseguire nella programmazione ed elaborazione dei vostri dati, potrete sempre contare sull'utilissimo e fondamentale apporto della schermata di aiuto, richiamabile con il tasto F3, che vi risolverà ogni genere di problemi inerenti al programma.

Giampiero Maietti

Grandi così

Caratteri giganti maneggevoli come quelli standard: reverse, lampeggio, multicolor, rotazione, inversione... e tutto alla velocità del linguaggio macchina, ed esportabile in qualsiasi programma Basic.

Big Print è una routine scritta completamente in linguaggio macchina che stampa caratteri ingranditi: 4 posizioni orizzontali e 4 verticali, per un totale di 16 byte di occupazione (**figura 1**).

Il bello di questo programma è che può essere agevolmente inserito in un qualsiasi programma Basic, consentendo di manipolare questi caratteri con la stessa facilità con cui il Basic standard tratta il set di default, anzi con qualche facility in più.

Big Print viene attivata per mezzo di una normale istruzione SYS, seguita però da 5 parametri oltre all'indirizzo di lancio. La sintassi quindi risulta essere: SYS M1,S,V,H,B,E. Analizziamo per prima cosa il significato dei parametri:

- **M1** è semplicemente l'indirizzo di impianto e deve sempre valere 49152.
- **S** è la stringa di caratteri che deve essere visualizzata. Può essere una costante come "MARCO" o "12 MAGGIO" oppure una variabile di tipo stringa come A$, V$() o qualsiasi altra espressione accettata dal Basic che restituisce come risultato una stringa di caratteri.
- **V** è il numero di linea da cui deve essere visualizzata la stringa e deve essere compreso tra 0 e 21.
- **H** è la posizione orizzontale e può variare da 0 a 36. Anche in questo caso per entrambi i parametri è accettabile qualsiasi tipo di espressione considerata valida dall'interprete Basic. E' anche possibile fare in modo che venga assunto come punto di stampa quello della posizione del cursore; in questo caso è sufficiente sostituire il parametro relativo con il simbolo "*" (asterisco).
- **B** indica il numero di spazi che devono separare i caratteri che compongono la scritta. Deve essere un valore compreso tra 0 e 32.
- **E** serve per specificare una serie di effetti speciali che possono essere introdotti nel messaggio da visualizzare. Sono accettati sia valori numerici sia stringhe.

Uso dei parametri

Può essere conveniente poter omettere alcuni parametri; in questi casi è necessario indicare la presenza di un parametro utilizzando il separatore (la virgola) e il valore di default sarà 0. Per esempio la chiamata SYS M1,"CIAO",10,,0,0 è del tutto analoga a SYS M1,"CIAO",10,0,0,0.

Una forma ancora più sintetica dello stesso comando è SYS M1,"CIAO",10 in cui tutti i parametri mancanti dopo il 10 sono stati automaticamente settati a 0.

Qualora fosse necessario eseguire due o più chiamate successive è possibile farlo in modo molto compatto utilizzando il ; (punto e virgola). Per esempio la linea SYS M1,"CIAO",6,10: SYS M1,"MARCO",11,10 può essere sintetizzata con SYS M1,"CIAO",6,10;"MARCO",11,10

I due metodi producono lo stesso risultato, ma il secondo è più veloce.

Colore e settaggio

Il colore dei messaggi viene definito attraverso il colore corrente del testo, il cui codice è contenuto nella locazione 646. Quando viene eseguita la SYS automaticamente il colore di default sarà quello di linea.

Tuttavia è possibile includere un co-

Figura 1.

Listato

```
$c000 lda $d1
$c002 pha
$c003 lda $d2
$c005 pha
$c006 jsr $aefd
$c009 jsr $ad9e
$c00c bit $0d
$c00e bmi $c015
$c010 ldx #$16
$c012 jmp $a437
$c015 ldy #$00
$c017 lda ($64),y
$c019 bne $c021
$c01b jsr $a8f8
$c01e jmp $c0cf
$c021 sta $d7
$c023 iny
$c024 lda ($64),y
$c026 sta $fb
$c028 iny
$c029 lda ($64),y
$c02b sta $fc
$c02d lda $64
$c02f ldy $65
$c031 jsr $b6db
$c034 jsr $c105
$c037 bcc $c03d
$c039 lda $d6
$c03b bcs $c043
$c03d lda #$16
$c03f jsr $c0d6
$c042 txa
$c043 pha
$c044 jsr $c105
$c047 bcc $c05e
$c049 lda $d3
$c04b tax
$c04c cmp #$28
$c04e bcc $c055
$c050 sbc #$28
$c052 clv
$c053 bvc $c04b
$c055 lda #$25
$c057 sta $ff
$c059 jsr $c0f6
$c05c bcc $c063
$c05e lda #$25
$c060 jsr $c0d6
$c063 stx $ff
$c065 pla
$c066 tax
$c067 jsr $e9f0
$c06a clc
$c06b lda $d1
$c06d adc $ff
$c06f sta $d1
$c071 bcc $c075
$c073 inc $d2
$c075 jsr $ea24
$c078 lda #$21
$c07a jsr $c0d6
$c07d stx $ce
$c07f jsr $c115
$c082 bcc $c0b9
$c084 bne $c093
$c086 lda #$00
$c088 sta $9e
$c08a jsr $c124
$c08d jsr $0073
$c090 jmp $c009
$c093 jsr $aefd
$c096 jsr $ad9e
$c099 bit $0d
$c09b bmi $c0a5
$c09d jsr $b7a1
$c0a0 stx $9e
$c0a2 jmp $c0bb
$c0a5 ldy #$00
$c0a7 lda ($64),y
$c0a9 beq $c0b9
$c0ab iny
$c0ac lda ($64),y
$c0ae sta $fd
$c0b0 iny
$c0b1 lda ($64),y
$c0b3 sta $fe
$c0b5 ldy #$00
$c0b7 lda ($fd),y
$c0b9 sta $9e
$c0bb lda $64
$c0bd ldy $65
$c0bf jsr $b6db
$c0c2 jsr $c115
$c0c5 bcc $c0c9
$c0c7 beq $c08a
$c0c9 jsr $a8f8
$c0cc jsr $c124
$c0cf pla
$c0d0 sta $d2
$c0d2 pla
$c0d3 sta $d1
$c0d5 rts
$c0d6 sta $ff
$c0d8 jsr $c115
$c0db bcc $c0e9
$c0dd beq $c0e9
$c0df jsr $aefd
$c0e2 jsr $0079
$c0e5 cmp #$2c
$c0e7 bne $c0ec
$c0e9 ldx #$00
$c0eb rts
$c0ec jsr $ad9e
$c0ef bit $0d
$c0f1 bmi $c100
$c0f3 jsr $b7a1
$c0f6 cpx $ff
$c0f8 bcc $c0ff
$c0fa ldx #$0e
$c0fc jmp $a437
$c0ff rts
$c100 ldx #$16
$c102 jmp $a437
$c105 ldy #$01
$c107 lda ($7a),y
$c109 cmp #$ac
$c10b beq $c10f
$c10d clc
$c10e rts
$c10f jsr $aefd
$c112 jmp $0073
$c115 jsr $0079
$c118 beq $c122
$c11a eor #$3a
$c11c beq $c122
$c11e cmp #$01
$c120 sec
$c121 rts
$c122 clc
$c123 rts
$c124 lda #$00
$c126 sta $bf
$c128 ldx $0286
$c12b stx $6e
$c12d stx $6f
$c12f sta $fd
$c131 lda $d018
$c134 and #$02
$c136 asl
$c137 asl
$c138 ora #$d0
$c13a sta $fe
$c13c lda $dc0e
$c13f and #$fe
$c141 sta $dc0e
$c144 ldy #$ff
$c146 sty $69
$c148 iny
$c149 sty $cf
$c14b sty $be
$c14d lda $ce
$c14f jsr $c317
$c152 bit $d011
$c155 bpl $c152
$c157 ldy $69
$c159 iny
$c15a sty $69
$c15c lda ($fb),y
$c15e cmp $9e
$c160 bne $c16b
$c162 lda $bf
$c164 eor #$80
$c166 sta $bf
$c168 jmp $c1e1
$c16b clv
$c16c cmp #$12
$c16e beq $c1a5
$c170 cmp #$92
$c172 beq $c1ab
$c174 cmp #$0e
$c176 beq $c1b1
$c178 cmp #$8e
$c17a beq $c1b5
$c17c cmp #$91
$c17e beq $c1bb
$c180 cmp #$11
$c182 beq $c1bf
$c184 cmp #$1d
$c186 beq $c1c3
$c188 cmp #$9d
$c18a beq $c1ca
$c18c jsr $c331
$c18f bpl $c1e1
$c191 bit $bf
$c193 bpl $c1eb
$c195 cmp #$c1
$c197 beq $c1d1
$c199 ldx #$08
$c19b cmp $c36f,x
$c19e beq $c1d7
$c1a0 dex
$c1a1 bpl $c19b
$c1a3 bmi $c1eb
$c1a5 lda #$80
$c1a7 sta $cf
$c1a9 bvc $c1e1
$c1ab lda #$00
$c1ad sta $cf
$c1af bvc $c1e1
$c1b1 lda #$d8
$c1b3 bvc $c1b7
$c1b5 lda #$d0
$c1b7 sta $fe
$c1b9 bvc $c1e1
$c1bb lda #$28
$c1bd bvc $c1cc
$c1bf lda #$28
$c1c1 bvc $c1c5
$c1c3 lda #$01
$c1c5 jsr $c2fd
$c1c8 bvc $c1e1
$c1ca lda #$01
$c1cc jsr $c317
$c1cf bvc $c1e1
$c1d1 lda #$01
$c1d3 sta $70
$c1d5 bne $c1e1
$c1d7 lda $bf
$c1d9 ora $c378,x
$c1dc and $c381,x
$c1df sta $bf
$c1e1 dec $d7
$c1e3 bne $c1e8
$c1e5 jmp $c2f4
$c1e8 jmp $c157
$c1eb ldx #$07
$c1ed cmp $c34f,x
$c1f0 bcs $c1f5
$c1f2 dex
$c1f3 bne $c1ed
$c1f5 eor $c357,x
$c1f8 ora $cf
$c1fa ldx #$00
$c1fc stx $6b
$c1fe asl
$c1ff rol $6b
$c201 asl
$c202 rol $6b
$c204 asl
$c205 rol $6b
$c207 adc $fd
$c209 sta $6a
$c20b lda $6b
$c20d adc $fe
$c20f sta $6b
$c211 lda $01
$c213 and #$fb
$c215 sta $01
$c217 ldy #$07
$c219 lda ($6a),y
$c21b sta $0061,y
$c21e dey
$c21f bpl $c219
$c221 bit $bf
$c223 bvc $c239
$c225 ldx #$00
$c227 ldy #$03
$c229 lda $61,x
$c22b pha
$c22c lda $0065,y
$c22f sta $61,x
$c231 pla
$c232 sta $0065,y
$c235 inx
$c236 dey
$c237 bpl $c229
$c239 lda #$20
$c23b bit $bf
$c23d beq $c24e
$c23f ldx #$07
$c241 ldy #$08
$c243 asl $61,x
$c245 ror
$c246 dey
$c247 bne $c243
```

(Continua)

```
$c249 sta $61,x
$c24b dex
$c24c bpl $c241
$c24e lda #$10
$c250 bit $bf
$c252 beq $c257
$c254 clv
$c255 bvc $c25e
$c257 lsr
$c258 bit $bf
$c25a beq $c27b
$c25c adc #$7f
$c25e ldy #$07
$c260 ldx #$07
$c262 bvs $c269
$c264 lsr $61,x
$c266 ror
$c267 bvc $c26c
$c269 asl $61,x
$c26b rol
$c26c dex
$c26d bpl $c262
$c26f pha
$c270 dey
$c271 bpl $c260
$c273 ldx #$07
$c275 pla
$c276 sta $61,x
$c278 dex
$c279 bpl $c275
$c27b lda $01
$c27d ora #$04
$c27f sta $01
$c281 lda #$00
$c283 sta $6c
$c285 clc
$c286 lda $be
$c288 adc $ce
$c28a tay
$c28b lda #$04
$c28d sta $6d
$c28f ldx $6c
$c291 lda #$00
$c293 rol $61,x
$c295 rol
$c296 rol $61,x
$c298 rol
$c299 rol $62,x
$c29b rol
$c29c rol $62,x
$c29e rol
$c29f sta $ff
$c2a1 lda $bf
$c2a3 and #$04
$c2a5 beq $c2ba
$c2a7 ldx $ff
$c2a9 beq $c2c6
$c2ab lda ($d1),y
$c2ad ldx #$0f
$c2af cmp $c35f,x
$c2b2 beq $c2b9
$c2b4 dex
$c2b5 bpl $c2af
$c2b7 bmi $c2c6
$c2b9 txa
$c2ba ora $ff
$c2bc tax
$c2bd lda $c35f,x
$c2c0 sta ($d1),y
$c2c2 lda $6e
$c2c4 sta ($f3),y
$c2c6 iny
$c2c7 dec $6d
$c2c9 beq $c2cd
$c2cb bne $c28f
$c2cd ldx $6c
$c2cf cpx #$06
$c2d1 beq $c2df
$c2d3 inx
$c2d4 inx
$c2d5 stx $6c
$c2d7 tya
$c2d8 clc
$c2d9 adc #$24
$c2db tay
$c2dc jmp $c28b
$c2df dec $d7
$c2e1 beq $c2f4
$c2e3 tya
$c2e4 sec
$c2e5 sbc #$78
$c2e7 sta $be
$c2e9 lda $6e
$c2eb ldx $6f
$c2ed sta $6f
$c2ef stx $6e
$c2f1 jmp $c157
$c2f4 lda $dc0e
$c2f7 ora #$01
$c2f9 sta $dc0e
$c2fc rts
$c2fd sta $ff
$c2ff clc
$c300 lda $d1
$c302 adc $ff
$c304 sta $d1
$c306 bcc $c30b
$c308 inc $d2
$c30a clc
$c30b lda $f3
$c30d adc $ff
$c30f sta $f3
$c311 bcc $c315
$c313 inc $f4
$c315 clv
$c316 rts
$c317 sta $ff
$c319 sec
$c31a lda $d1
$c31c sbc $ff
$c31e sta $d1
$c320 bcs $c325
$c322 dec $d2
$c324 sec
$c325 lda $f3
$c327 sbc $ff
$c329 sta $f3
$c32b bcs $c32f
$c32d dec $f4
$c32f clv
$c330 rts
$c331 ldx #$0f
$c333 cmp $e8da,x
$c336 bne $c34b
$c338 txa
$c339 ldx $70
$c33b beq $c346
$c33d sta $6d,x
$c33f txa
$c340 asl
$c341 and #$03
$c343 sta $70
$c345 rts
$c346 sta $6e
$c348 sta $6f
$c34a rts
$c34b dex
$c34c bpl $c333
$c34e rts
$c34f brk
$c350 jsr $6040
$c353 ---
$c354 ldy #$c0
$c356 ---
$c357 ---
$c358 brk
$c359 rti
$c35a jsr $c040
$c35d ---
$c35e lda ($20,x)
$c360 jmp $(627b)
$c363 ---
$c364 sbc ($ff,x)
$c366 inc $7f7e,x
$c369 adc ($fc,x)
$c36b ---
$c36c ---
$c36d cpx $d5a0
$c370 eor $c9,x
$c372 eor #$cc
$c374 ---
$c375 ---
$c376 ---
$c377 ---
$c378 rti
$c379 brk
$c37a jsr $1000
$c37d php
$c37e ---
$c37f brk
$c380 brk
$c381 ---
$c382 ---
$c383 ---
$c384 ---
$c385 ---
$c386 ---
$c387 ---
$c388 ---
$c389 brk
```

lore diverso introducendo direttamente il carattere di controllo nella stringa da visualizzare, come nelle usuali PRINT.

Un discorso analogo vale anche per il set di caratteri (maiuscolo/minuscolo): infatti per default il messaggio viene visualizzato con lo stesso set impostato dal sistema operativo, tuttavia inglobando il carattere di controllo del set alternativo direttamente nella stringa è possibile cambiarlo.

Va notato che per quanto riguarda sia il colore sia il set, i cambiamenti introdotti dai caratteri di controllo presenti nella stringa sono locali alla chiamata e vengono perduti all'uscita.

Similmente includendo nella stringa caratteri di controllo per il reverse o per il posizionamento del cursore si ottiene l'effetto desiderato analogo a quello delle normali PRINT, tuttavia al ritorno al Basic vengono sempre ripristinate le condizioni preesistenti alla chiamata, sia per la condizione di reverse sia per la posizione del cursore.

Effetti speciali

Abbiamo visto come manipolare questi caratteri giganti per ottenere messaggi che, a parte le dimensioni, sono del tutto analoghi a quelli delle normali PRINT. Ora invece analizzeremo le possibilità offerte da BigPrint di ottenere effetti speciali.

L'ultimo parametro della SYS, chiamato E, serve appunto a indicare alla routine che gestisce le scritte che nella stringa possono essere presenti caratteri di controllo per generare effetti speciali e che quindi non vanno interpretati come testo.

Questo parametro non è altro che un carattere ASCII che delimita la sequenza di caratteri di controllo all'interno della stringa da stampare e funziona più o meno come le virgolette per il tradizionale quote mode dell'editor. Infatti i caratteri della stringa compresi tra le prime due occorrenze del carattere di controllo E vengono trattati come indicazioni per effetti speciali. Con riferimento alla **tavola 1** per i codici degli effetti speciali, vediamo alcuni esempi:

```
SYS49152,"/U/ciao",10,10,2,"/"
SYS49152,"/R/ciao",10,10,1,47
```

In entrambi i casi il carattere per gli effetti speciali è il simbolo di divisione "/": nel primo caso è stato indicato esplicitamente, nel secondo per mezzo del suo codice ASCII che è appunto 47.

Il primo esempio visualizza la paro-

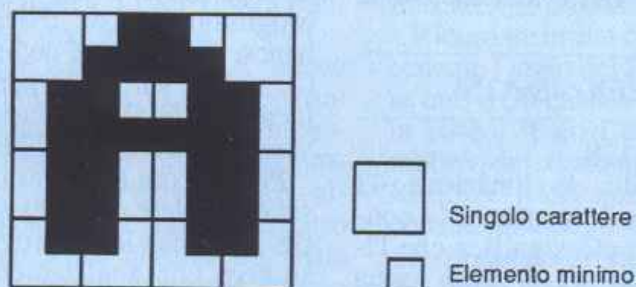
Tavola 1. Sommario dei codici di controllo

Codici riconosciuti in ogni caso

Carattere	ASCII	Effetto
Reverse on	18	Attiva il reverse mode
Reverse off	146	Disattiva il reverse mode
Lower case	14	Attiva il set minus/maius
Upper case	142	Attiva il set maius/grafici
Cursor up	145	Sposta in alto di una riga
Cursor down	17	Sposta in basso di una riga
Cursor left	157	A sinistra di una colonna
Cursor right	29	A destra di una colonna
Codici colore	-	Cambiano il colore di linea

Codici riconosciuti solo nel Control Mode

Carattere	ASCII	Effetto
Maius A	193	Alterna due colori nella stringa
Maius U	213	Attiva il modo capovolto
Minus u	85	Disattiva il modo capovolto
Maius I	201	Attiva il modo inverso
Minus i	73	Disattiva il modo inverso
Maius L	204	Ruota a sinistra
Maius R	210	Ruota a destra
Maius O	207	Attiva il modo overlay
Minus o	79	Disattiva il modo overlay
Maius C	195	Ripristina il default

Tavola 2. Definizione dei caratteri giganti

la ciao alle coordinate 10,10 con 2 spazi tra ogni lettera e con le lettere capovolte (effetto U). Il secondo visualizza la stessa stringa, a partire dalla stessa posizione, ma con un solo carattere di separazione e con il messaggio ruotato verso destra.

Va sottolineato che gli effetti speciali sono cumulativi e valgono per tutta la chiamata, compreso l'eventuale collegamento con ";".

Per questo motivo è stato inserito un particolare codice che consente di azzerare in un colpo tutti gli effetti.

Ecco una descrizione dei codici:

- **Shift A**: Seguito dai codici di due colori fa in modo che il messaggio che segue venga stampato con i caratteri di colore alternato.
- **Shift U:** Capovolge tutti i caratteri del messaggio.
- **Minus u:** Disattiva la condizione precedente.
- **Shift I:** Attiva l'inversione dei caratteri.
- **Minus i:** Disattiva la condizione precedente.
- **Shift L:** Ruota la scritta verso sinistra.
- **Shift O:** Attiva il modo di overlay, cioè di sovrapposizione. Infatti nel modo normale se due messaggi si sovrappongono solo l'ultimo viene visualizzato, mentre in questo modo viene rappresentata la composizione dei pattern che definiscono i caratteri (**figura 2**).
- **Minus o:** Disattiva la condizione precedente.
- **Shift C:** Azzera tutti gli effetti speciali e riporta alla condizione di default.

Considerazioni tecniche

BigPrint lavora direttamente sulla memoria di schermo e sulla memoria colore bypassando le routine di print del sistema operativo. Questo consente naturalmente una maggiore velocità di esecuzione e nello stesso tempo preserva i puntatori del Basic.

BigPrint infatti utilizza i suoi puntatori e non effettua controlli sulla posizione del cursore o sui caratteri della stringa da visualizzare per vedere se eccede i limiti della memoria di schermo o di colore.

Ciò vuol dire che gli unici messaggi di errore che vengono emessi sono dovuti all'assegnamento di valori illegali ai parametri della SYS (illegal quantity).

In alcuni casi per creare effetti speciali può essere utilizzata la routine del sistema operativo di sbiancamento della linea che parte da 59903.

E' sufficiente eseguire una POKE 781,L: SYS 59903 con L compreso tra 0 e 24, e la linea L verrà ripulita.

Quasi tutti i compilatori Basic trattano impropriamente l'istruzione SYS con parametri.

Quindi se volete compilare un programma che utilizzi BigPrint dovete leggere attentamente il manuale del vostro compilatore riguardo questo argomento.

In particolare con Blitz! è sufficiente far precedere ogni chiamata da due caratteri di due punti, per esempio:

```
100 ::SYS49152,"CIAO",10,5.
```

Nella cassetta allegata a questo numero troverete un file che contiene sia il codice macchina della routine sia un piccolo programma dimostrativo. Per ottenere il codice macchina puro (da $C000 a $C38A) è sufficiente selezionare l'opzione corrispondente nel meù principale e indicare il nome del file con cui si vuole salvarlo e il numero di device (8=disco, 1=nastro).

Roberto Russo

I segreti del Basic

Tutti sanno utilizzare il Basic. Pochi in realtà conoscono esattamente come funziona. Questo articolo farà un po' di luce sul modo in cui il Commodore 64 gestisce i programmi Basic.

In questo articolo vedremo in dettaglio come il Commodore 64 e il Commodore 128 immagazzinano i programmi Basic, e come utilizzare questa tecnica per ottenere utilissimi vantaggi.Vedremo come ottenere le REM del listato in reverse, come rendere il listato Basic più leggibile, come proteggere o sproteggere (senza approfondire eccessivamente) un programma interamente Basic, come nascondere alcune parti di programma agli occhi della routine di List, come ottenere un AutoRun permanente, come ottenere un Merge di due programmi Basic senza utilizzare il linguaggio macchina e altro ancora.

Locazioni e puntatori

I programmi Basic vengono automaticamente memorizzati a partire da una predeterminata locazione di memoria, evitando così il problema di decidere dove locare ciò che digitate (problema con cui si ha a che fare se si programma in linguaggio macchina), risparmiando così tempo e decisioni. Il programma Basic inizia dalla locazione numero 2049 sul C64 e dalla 7169 o 16385 sul C128. Questo significa che a partire da questa locazione verrà memorizzato tutto ciò che scriverete preceduto da un numero di linea (le istruzioni date in modo diretto non vengono ricordate).

Il motivo per cui il C128 dispone di due valori è molto semplice: se si sta usando la pagina grafica (data con il comando Graphic n, con n > 0>) l'inizio Basic sarà in 16385, mentre se non la si sta utilizzando, l'inizio Basic sarà in 7169. Ricordate a tale proposito che per abbassare l'inizio Basic in 7169 (sul C128) dovete utilizzare l'istruzione Graphic CLR, che farà perdere il contenuto della pagina grafica.

Vi sono due locazioni che puntano all'inizio del Basic. Queste due locazioni sono la 43 e la 44 sul C64 e la 45 e 46 sul C128. Ciò significa che se modificate il contenuto di queste locazioni, modificherete l'inizio dell'area Basic, trasferendola in un'altra zona.

Il Basic inizia alla locazione numero:

```
PEEK<43> + PEEK<44>*256
```

sul Commodore 64, e alla locazione numero:

```
PEEK>45> + PEEK<46>*256
```

sul Commodore 128.

Sul C64 infatti, la locazione 43 contiene il valore 1, mentre la 44 contiene il valore 8: ciò significa che l'inizio Basic è in <1+8*256> ossia 2049 come già detto.

D'ora in avanti ci riferiremo solo al C64: chi possiede un C128 deve ricordarsi di utilizzare i puntatori del suo computer e non quelli del C64 forniti negli esempi. La forma con cui un programma Basic viene posta in memoria è la seguente:

```
$ LINK (lb)
* LINK (hb)
* NUMERO DI LINEA (lb)
* NUMERO DI LINEA (hb)
* istruzioni Basic introdotte
* 0
* LINK successivo (lb)
* LINK successivo (hb), ecc.
```

Vedremo fra poco un esempio che chiarirà ogni dubbio in merito.

Allo stesso modo in cui, nel C64, le locazioni 43/44 puntano all'inizio del Basic, le locazioni 45/46 puntano alla fine. Ciò significa che un programma Basic sul C64 occupa un numero di locazioni così ottenibile:

```
PRINT <PEEK<<45> PEEK<46>
*256>
<PEEK<43>+PEEK<44>*256>
```

E' intuibile quindi che i puntatori 45/46 vengono modificati ogni volta che introducete, modificate o cancellate una linea di programma in memoria. Sul Commodore 128 non esistono i puntatori di fine Basic perché il programma Basic risiede in un suo banco di memoria dedicato (il banco 0): quindi sul C128 per conoscere la locazione finale del Basic occorre scrivere: PRINT 65279 - FRE <0>.

Provate ora a scrivere due linee Basic e a vedere come esse vengono poste in memoria; il programma dimostrativo che analizzeremo sarà il seguente:

```
10 PRINT"UNO"
20 PRINT"DUE"
```

Fate attenzione a non aggiungere uno spazio <CHR$>32>> dopo la PRINT. E' inutile dire che il programma è puramente dimostrativo. La **tavola 1** mostra il contenuto delle locazioni di memoria in cui sono memorizzate le due linee Basic.

La linea Basic 10 è posta in memoria a partire dalla locazione numero 2049. Ecco i significati delle varie locazioni: 2049 e 2050 contengono rispettivamente 12 e 8; abbiamo detto che le prime due locazioni rappresentano il link (ossia l'indirizzo da cui partirà la prossima linea Basic). Infatti 12 + 8*256 dà come risultato proprio 2060, ossia la locazione della linea Basic successiva. Il motivo per cui si moltiplica il secondo numero per 256

e lo si somma al primo è che un byte può contenere al massimo il valore 255: numeri maggiori vanno divisi in due parti (il byte basso LB e il byte alto HB), e questa è la formula per riottenere il numero originario. Le locazioni 2951 e 2052 contengono rispettivamente 10 e 0: questo è il numero di linea Basic diviso nella parte alta e bassa (10 +0*256 + 10). La 2053 contiene il valore 153 (vedere in **tavola 2** la tabella dei Token e dei codici Ascii): 153 è infatti il Token della istruzione Basic Print.

Salvare l'istruzione Print sotto forma Ascii sarebbe uno spreco di memoria troppo elevato, per cui ogni parola riservata Basic ha un suo Token, ossia un valore superiore a 128 con cui viene salvata in memoria, occupando così solo un byte e non cinque.

Dalla locazione 2054 compresa alla 2058 compresa troviamo ciò che si era scritto dopo la Print, ossia il contenuto "UNO" memorizzato sotto forma di codici Ascii. Infatti il codice Ascii del carattere " è 34, del carattere U è 85 e così via. A questo punto la linea 10 è terminata, e infatti la locazione 2059 contiene uno 0 (ossia 'fine linea'). Dopo questa cella inizia la seconda linea introdotta (in questo caso la 20) con le locazioni 2060 e 2061 che contengono il link puntante in 2071 >23 + 8*256 + 2071> e vedremo poi cosa indica lo zero extra puntato da queste due locazioni.

2062 e 2063 contengono il valore 20 (seconda linea Basic). 2064 contiene ancora il valore 153: è la seconda istruzione Print presente nel programma. Da 2965 a 2069 troviamo gli altri codici Ascii del secondo messaggio che stampiamo con la linea 20, ossia "DUE". In 2070 troviamo ancora uno 0 che indica la fine della linea 20. In 2071 e in 2072 vi sono altri due zeri: si tratta dei due zeri consecutivi che ci indicano la fine del programma Basic in memoria. Le locazioni 45/46 puntano infatti all'ultimo di questi due zeri ossia alla fine del programma Basic in memoria.

A questo punto si dà per scontato che sia chiaro come un programma Basic viene posto in memoria. Se così non fosse, il metodo migliore è quello di eseguire un paio di prove, con linee Basic molto corte, e poi di andare a verificare cosa è realmente presente in memoria digitando:

```
FORA+PEEK<43>+PEEK<44>
*256TOPEEK<<45>+PEEK
<46>*256:PRINTA,PEEK <A>
:NEXT (return)
```

ottenendo così l'elenco delle locazioni in cui è memorizzato il programma affiancate dal loro contenuto.

Tavola 1. Struttura di un programma Basic

Locazione	Contenuto	Carattere Ascii
2049	12 (link LB)	-
2050	8 (link HB)	-
2051	10 (linea LB)	-
2052	0 (linea HB)	-
2053	153 (print)	-
2054	34	"
2055	85	u
2056	78	n
2057	79	o
2058	34	"
2059	0 (fine linea)	-
2060	23 (link LB)	-
2061	8 (link HB)	-
2062	20 (linea LB)	-
2063	0 (linea HB)	-
2064	153 (print)	-
2065	34	"
2066	68	d
2067	85	u
2068	69	e
2069	34	"
2070	0 (fine linea)	-
2071	0 (fine prg)	-
2072	0 (fine prg)	-

Ricordate infine che la locazione precedente l'inzio del Basic deve contenere uno 0 (in questo caso la locazione è la 2048): se così non fosse, comparirebbero dei messaggi di SYNTAX ERROR e il RUN non verrebbe preso in considerazione. E' quindi importantissimo quando si va a spostare l'inizio del Basic che la locazione precedente l'inizio venga posta a 0: la formula generica che lo permette è:

```
POKE<PEEKL<43>+PEEK<44>
*256>-1,0 (return) (45/46 sul C128)
```

Ecco quindi come ingannare il tipo più banale di AutoRun: basta digitare in modo diretto: POKE 2048,1 (return), e caricare il programma: l'autorun dato servirà solo a fornire un ?SINTAX ERROR e a ridarci il controllo. Basta ora rimettere uno 0 nella locazione 2048 e tutto è a posto.

Prime applicazioni

Iniziamo a vedere le prime applicazioni ottenibili dopo le premesse date.

1. Modifica dell'inizio Basic.

Le applicazioni da cui si trarrà vantaggio imparando a utilizzare questa tecnica sono talmente numerose che non è possibile neppure un sommario elenco. Proviamo ora come applicazione a modificare l'inizio del Basic alle precedenti due linee che andranno copiate senza alcuna modifica. Si è già dimostrato come la seconda linea (la 20), inizi dalla locazione numero 2060. Digitate ora in modo diretto

```
POKE43,PEEK<2049>:POKE44,PE-
EK<2050> <R>
```

dove <R> indica la pressione del tasto Return. Come noterete, chiedendo il listato, la linea 10 non esiste più (provate a dare un RUN per accertarvene). Il motivo è semplicissimo: con la precedente istruzione siamo andati a leggere il primo link (quello della linea 10) il quale punta alla seconda linea e a tali valori abbiamo posto l'inizio del Basic: in tal modo questo link contiene già i due valori (LB e HB) in cui dobbiamo trasferire l'inizio del Basic, risparmiandoci così tutti i calcoli sulla divisione in LB e HB. Se non salvassimo il programma a questo punto, salveremmo la sola linea 20, in quanto la 10, pur essendo in memoria a tutti gli effetti, non viene considerata dal computer come linea Basic. Digitate ora in modo diretto:POKE43,1:POKE44,8 e la linea 10 ricompare nuovamente (eseguibile, listabile e

salvabile). Alcune prove vi permetteranno di comprendere bene questa tecnica. La prima applicazione pratica è la seguente: se dotate un programma di autorun (si vedrà più avanti come fare) potete modificare l'inizio Basic dall'interno del programma che conterrà delle linee essenziali, in modo da escludere dall'area Basic il numero di linee desiderato. Un salvataggio del programma dato a questo punto salverebbe solo una fetta di programma e non tutto il listato, rendendo inutilizzabile il salvataggio eseguito.

2. Modifica dei link.

A differenza della modifica dell'inizio del Basic, la modifica dei link ha effetto solo sul listato del programma e non sull'esecuzione dello stesso. Provate a resettare il computer e a ridigitare sempre le due linee Basic precedenti e a scrivere in modo diretto:

```
POKE2049,PEEK<2060>:POKE
2050,PEEK<2061> (return)
```

Chiedete ora il listato: noterete che la linea 20 è scomparsa. Provate poi a dare il RUN: vedrete come la linea 20, anche se non letta dalla routine di LIST, viene eseguita regolarmente. Cosa abbiamo modificato? Con la poke precedente, abbiamo fatto puntare il primo link <2049/2050> al valore contenuto nel secondo link <2060 /2061>; in tal modo il primo link (della linea 10), punta direttamente alla fine del programma Basic, ma lo zero contenuto in 2059 informa sempre che la linea 10 termina a quel punto, e la routine di list crede che il programma sia terminato (il link successivo, infatti, lo porta a leggere i due zeri indicanti la fine del programma). Le applicazioni di questa tecnica sono decisamente interessanti: se fate puntare un link di linea anche all'interno di un programma a un altro link situato più avanti, tutte le linee intermedie non verranno più lette dalla routine di list, e anche chi conosce questi trucchi desisterà ben presto dal rimettere le cose a posto, dato l'eccessivo numero di controlli da eseguire se il programma è almeno di media lunghezza. Questo tipo di protezione parziale è una protezione statica: se salvate un programma con i link modificati, il programma verrà salvato ugualmente (o caricato) ma le linee nascoste continueranno a essere tali.

3. Merge di due programmi.

L'operazione di Merge consiste nell'unione grezza di due programmi, uno residente in memoria e l'altro su disco o cassetta. Date le premesse fatte, siamo ora in grado di compiere il modo velocissimo questa operazione in modo manuale e senza avere più bisogno di alcun programma in linguaggio macchina. Abbiamo detto che un programma Basic termina con due zeri indicanti appunto la fine del programma, e che l'ultimo di questi zeri è puntato da 45/46. Abbiamo visto inoltre come l'ultimo link dell'ultima linea Basic ci fornisca direttamente la locazione da cui dovremmo caricare il secondo programma (in coda a uno già presente in memoria). Resta l'ultimo problema: quanto dista il link finale (quello da cui dovremo prelevare i valori) dalla fine del Basic? Un minimo di ragionamento ci mostra come tale valore vari col variare dell'ultima linea Basic in memoria. Ecco l'idea: se aggiungiamo in coda al programma in memoria sempre la stessa linea Basic, il link in questione disterà sempre lo stesso numero di locazioni dalla fine del Basic e potremo così scrivere una regola generale applicabile sempre. Ecco come procedere per punti:

- aggiungete in coda al programma Basic in memoria una linea nuova contenente una sola REM (es.40000 REM).
- scrivete in modo diretto:PRINT PEEK<45> + PEEK <46>*256 (return) e prendete nota del numero che verrà fornito (lo chiameremo X).
- scrivete sempre in modo diretto: POKE43,PEEK<X-8> (return) e POKE44,PEEK<X-7> (return), dove X è il valore fornito al punto precedente.
- caricate ora il secondo programma Basic con un normale LOAD.
- digitate in modo corretto: POKE43,1 e POKE44,8.

I due programmi Basic sono ora uniti in memoria e consecutivamente, e possono essere eseguiti, listati e salvati. Come tutti i programmi di Merge è naturalmente richiesto che il primo numero di linea del secondo programma sia superiore all'ultimo numero di linea del primo programma in memoria, e che i due programmi possano essere entrambi contenuti in memoria senza generare degli out of memory.

Il funzionamento dovrebbe essere chiaro: introducendo la REM del punto uno abbiamo modo di sapere dove si trova l'ultimo link che ci fornisce la locazione da cui caricare il secondo programma per fare in modo che sia linkato correttamente. Il secondo punto preleva proprio questi due valori e li pone direttamente in 43/44, ossia alza l'inizio del Basic al punto da cui si deve caricare il programma B. Ora il LOAD carica il secondo programma proprio da questa locazione (correttamente linkata) e le ultime due poke rimettono l'inizio Basic ai suoi abitudinari valori (ossia 2049).

Inutile soffermarsi sull'utilità di questa routine: una volta imparata a memoria consente in un attimo di unire due programmi Basic senza perdere tempo.

Un consiglio:se create una enciclopedia di routine Basic per i vostri programmi, la salvate su periferica e la numerate con un numero di linee altissimo, potete usare il precedente Merge per caricare la routine desiderata ogni volta che lo volete. Occorrerà poi avere sempre in memoria un programma in L/M che risieda fuori dal Basic per renumerare tutto il programma ogni volta che avete caricato una routine da periferica. Potete usarlo anche per vedere la Directory senza perdere il programma: basterà in tal caso dare un New, prima di riabbassare l'inizio Basic in 2049.

Ricordate che ci sono alcune forme di caricamento della Direcotory che non tutti conoscono:

- LOAD"$*=PRO", che carica la Directory dei soli file programma,
- LOAD"$*=SEQ", che carica solo i file sequenziali ecc.
- Valide anche USR e REL, e la sostituzione dell'asterisco con una chiave di ricerca.

4. Reverse e colorazione.

Osservando la **tavola 2** noterete come alcuni codici presenti non corrispondano né a una lettera né a un Token, ma diano ugualmente degli effetti interessanti. Digitate il programma esempio:

```
10 REM*
20 PRINT "CIRO"
```

e scrivete in modo diretto: POKE 2054,5 (return). La seconda linea del listato apparirà sempre di colore bianco e l'asterisco dopo la REM è sparito. Cos'è successo? Semplicemente è stato creato uno spazio extra in memoria (l'asterisco) e manualmente è stato sostituito con il valore 5 (ossia il colore bianco). Provate a premere STOP+RESTORE e a scrivere PRINT CHR$<5> per sincerarvene. Il valore 5 posto dopo una REM viene in realtà eseguito, e non mostrato come una parte del LIST. Il programma infatti ignora la presenza del 5 in memoria, perché la REM lo fa passare all'esecuzione della linea successiva, mentre la routine di LIST lo deve leggere per forza, e leggendolo lo esegue. I codici che possiamo utilizzare per esperi-

menti di questo tipo sono i seguenti:

valore effetto

----

5. colora di bianco
8. disabilità SHIFT/COMMODOR-
...riabilita SHIFT/COMMODORE
13. cursore a capo
14. set minuscolo/maiuscolo
17. cursore giù
18. reverse on
28. colora di rosso
30. colora di verde
31. colora di blu
204. stampa SINTAX ERROR

Volendo quindi per esempio ottenere tutte le Rem del programma scritte in Reverse, basta sostituire il primo carattere seguente la Rem con un 18: l'effetto termina con la fine della linea su cui si pone il reverse (a differenza dei colori che restano attivi). Una routine che esegue l'operazione manualmente (empirica) è la seguente:

```
60000 I=PEEK<43>+PEEK<44>
*256
60010 F+PEEK<45>+PEEK<46>
*256
60020 FORA=1TOF
60030IFPEEK<A>=143 AND PEEK
<A+1>+32THENPOKEA+2,18
60040 NEXT
```

Abbiamo detto che è un modo empirico di procedere per il seguente motivo: il programmino (da aggiungere in coda al programma che si desidera e poi da cancellare) testa la presenza di un 143 (token della REM) e se è seguito da uno spazio (32) allora esegue la modifica. Ma un 143 seguito da un 32 può trovarsi nel programma anche per altri motivi (parti di un link o altro) e quindi la routine non è infallibile. Per accertarsene occorre vedere il listato: se tutto è OK, allora siamo a posto, se compaiono caratteri strani è necessario eseguire l'operazione manualmente (funziona in un buon 90% dei casi). Le modifiche date in questo modo restano permanentemente anche dopo il salvataggio e il successivo ricaricamento. Un valore 204 provoca invece, chiedendo il LIST, la stampa di un messaggio di errore e l'interruzione dell'esecuzione.

Se possedete un C128 (in modo 128) astenetevi da questi esperimenti: il C128 possiede oltre 140 comandi Basic e molti Token sono memorizzati nella parte bassa seguita dalla parte alta, per cui sorgono problemi inaspettati da tutte le parti; potete invece utilizzare il metodo della sostituzione manuale ogni volta che lo volete. La tecnica della sostituzione di caratteri dopo una Rem è utilizzata anche da chi vuole inserire cortissime routine in LM inglobandole all'interno del programma Basic.

Tavola 2.

Codici token del Basic

Token	Codice Ascii	Token	Codice Ascii
abs	182	or	176
and	175	peek	194
asc	198	poke	151
atn	193	pos	185
chr$	199	print	153
close	160	print#	152
clr	156	read	135
cmd	157	rem	143
cont	154	restore	140
cos	190	return	142
data	131	right$	201
def	150	rnd	187
dim	134	run	138
end	128	save	148
exp	189	sgn	180
fn	165	sin	191
for	129	spc	166
fre	184	sqr	186
get	161	status	---
get#	---	step	169
gosub	141	stop	144
goto	137	str$	196
if	139	sys	158
input	133	tab	163
input#	132	tan	192
int	181	then	167
left$	200	time	---
len	195	time$	---
let	136	to	164
list	155	usr	183
load	147	val	197
log	188	verify	149
mid$	202	wait	146
new	162	+	170
next	130	-	171
not	168	*	172
on	145	:	173
open	159	=	178

5.Protezioni e sprotezioni.

Per protezione di un programma si intende la non accessibilità al listato. Chi è in grado di programmare in linguaggio macchina (ed è in grado di redigere un intero programma in LM senza il Basic) è già in grado di inventarsi da solo sistemi per proteggere il suo lavoro. I problemi sorgono volendo proteggere un listato Basic dagli occhi dei copiatori di programmi e dai ladri di routine. Un sistema valido è la modifica dei link all'interno del programma, già spiegata in questo articolo. Un altro sistema adottato (ma nessuno vieta di usarli anche tutti contemporaneamente) è quello di dotare il programma di AutoRun e disabilitare il tasti STOP e RESTORE. Sugli AutoRun si sono visti in giro programmi validi affiancati da altri a dir poco ridicoli. Daremo qui alcune idee su come fornire un AutoRun da disco un po' valido (gli AutoRun su cassetta sono laboriosi da eseguire a mano e quindi è meglio procurarsi un programma protettore e vedere come funziona per costruirsene un altro migliore). Per AutoRun si intende la partenza del programma appena caricato, in maniera automatica.

Si distinguono due tipi di AutoRun:
1. $ AutoRun a riempimento di buffer.
2. * AutoRun immediati.

I secondi sono i più efficaci e vedremo come sia facile costruirsene uno senza bisogno del linguaggio macchina e in pochi secondi. Gli AutoRun a riempimento di buffer, come dice il nome, funzionano mandando in esecuzione un programmino in linguaggio macchina che riempie il buffer di tastiera con le lettere R-U-N-CHR$<13> e poi torna al Basic. Tornati al Basic viene scritto immediatamente RUN (ret.) e il programma parte. La routine in LM è la seguente:

```
CLI
LDA #4
STA 198
LDA #'R
STA 631
LDA #'U
STA 632
LDA #'N
STA 633
LDA #13
STA 634
SEI
RTS
```

I numeri sono espressi in decimale, con 'R si intende il codice ASCII della lettera R e così via. Andando a locare questa routine da 828 (buffer del registratore) e abbassando l'inizio del Basic in 770 (ove è contenuta la routine di READY) con POKE43,2 e POKE44,3, il primo dei due passi è compiuto. Ora dovete dividere il valore 828 nella parte alta e bassa secondo la formula:

```
H+INT<X/256>
L+X-H*256
```

dove H è la parte alta, L la parte bassa, e X il numero in questione (in questo caso 828). Ora dovete scrivere in modo diretto:POKE770,L:POKE771,H:SAVE"[clr]nome programma",8 (return). Terminato il salvataggio basta riportare il Basic ai soliti valori con POKE43,1 e POKE44,8. Nella formula appena data L rappresenta la parte bassa del numero, mentre H è la parte alta.

Perché funziona questo AutoRun? La soluzione è facile come sempre: la routine di READY (quella che viene eseguita a partire dalla locazione puntata da 770/771 ogni volta che vediamo il READY sullo schermo) può essere modificata a piacere e fatta puntare a ciò che desideriamo (purché in linguaggio macchina). Se la routine di READY, tramite modifica in 770/771 viene fatta puntare al programmino che riempie il buffer di tastiera con le lettere RUN (return) si ha la stampa del messaggio RUN (return) appena terminato il caricamento del programma. Infatti, quando caricate un programma, al termine vedete proprio la scritta READY. Il RUN impartito fa quindi partire il programma.

Abbiamo abbassato l'inizio del Basic in 770 e 771 perché così facendo salviamo tutta la memoria a partire dalla locazione numero 770 fino alla locazione numero PEEK<45>+PEEK<46>*256 e quindi salviamo alche il programmino di autorun in LM, la memoria video (1024/2023) e tutto il programma Basic da 2049 in poi. Quando carichiamo il programma salvato con un LOAD"nome",8,1 andiamo a modificare di nuovo le locazioni 770 e 771 e le facciamo riportare alla routine di AutoRun e quindi il programma parte. E' importante caricare il programma utilizzando l'indirizzo secondario (,8,1) perché in caso contrario il caricamento avverrebbe a partire dall'area Basic (2049).

L'ultima cosa da chiarire è il nome che abbiamo dato al programma: perché abbiamo utilizzato il [clr] all'interno del nome del programma? Si è detto che anche lo schermo video viene salvato assieme al programma Basic, quindi se non facessimo in questo modo, verrebbero salvate anche le istruzioni che stiamo scrivendo sullo schermo, dato che lo schermo (da1024) resta visibile durante il caricamento. Questo espediente cancella il video durante l'ultima fase del salvataggio e lascia solo il nome del programma scritto sullo schermo in alto a sinistra. Una volta salvato il programma su disco, è opportuno cambiargli nome per evitare di vedere sempre quello sgradevole simbolo in reverse all'interno del nome. Per compiere tale operazione si deve usare la routine di RENAME, previa apertura del canale 15 (vedere il manuale del disk in proposito). Se tutto il procedimento non fosse ben chiaro, si consideri la differenza fra un LOAD"nome",8 e un LOAD"nome",8,1. Nel primo caso il programma viene caricato a partire dalla locazione puntata da 43/44, mentre nel secondo caso viene caricato a partire dalla locazione da cui è stato salvato. Ciò significa che assieme al programma salvato su disco, il computer salva anche un paio di locazioni che indicano da quale zona deve essere caricato il programma. Un programmino per leggere queste due locazioni è il seguente:

```
10 SYS65511:OPEN8,8,12,"nome
programma"
20 GET#*,A*:GET#8,B$:CLOSE8
30 I=ASC<A$+Z$>+ASC<B$+Z$>
*256:END
```

al termine la variabile I conterrà la locazione da cui il programma è stato salvato, e da cui verrà caricato con LOAD"nome",8,1. Se I vale 2049, è un programma Basic, in caso contrario è molto probabilmente un programma in LM, oppure un programma Basic salvato da un C128 o un programma Basic salvato dopo una modifica di puntatori.

Di natura analoga al precedente è il secondo tipo di AutoRun, quello permanente, molto più efficace del precedente.

Immaginate il metodo appena visto: a rendere scomoda la preparazione di questo AutoRun è l'inserimento del linguaggio macchina e il fatto che il programma non parte direttamente, ma viene prima eseguita una routine in LM che dà l'AutoRun. Se il programma potesse partire con una SYS di attivazione tutta l'operazione diventerebbe talmente veloce e immediata da sembrare quasi un gioco. In altre parole, immaginiamo ancora un attimo di avere un programma che parta con una SYS (e non con un RUN): basterebbe solo abbassare il Basic in 770 come nel programma precedente, e far puntare la routine di READY (in 770) all'inizio del programma che parte con una SYS e il gioco sarebbe fatto. Naturalmente, un programma che parte con una SYS deve essere interamente in linguaggio macchina, a meno che...non venga compilato.

Prendete un programma Basic a caso e compilatelo con un qualsiasi compilatore (PETSPEED, BLITZ o ciò che possedete). Caricate il programma compilato in memoria e abbassate l'inizio del Basic in 770, con una POKE 43,2 e POKE44,3. Dividete la SYS di attivazione del compilatore (quella che viene indicata nell'unica linea Basic che il compilatore lascia, e che è 2073 per PETSPEED, o un altro valore per gli altri compilatori) nella parte alta e bassa seconda la formula prima descritta, quindi salvate il programma con POKE770,L:POKE771,H:SAVE"[clr]nome",B. Resettate ora il computer e verificate il corretto funzionamento del programma appena salvato caricandolo con LOAD"?nome",8,1. Premendo RUN/STOP+RESTORE, la fuoriuscita dal programma non fa altro che farlo ripartire da capo, facendovi perdere per sempre il controllo diretto della tastiera e facendovi ritrovare di continuo all'interno del programma in esecuzione.

Fausto Molinari

Senza speranza

Una fantastica avventura nel cosmo alla ricerca di cuori dispersi. Niente guerre stellari e niente astronavi a disposizione: una tuta spaziale e un jet, per gustarsi meglio il panorama.

L'ultimo grido nel settore degli adventure consiste nell'unire, al semplice testo, delle schermate grafiche in grado di dare una chiara visione della situazione di gioco; anche se non tutti i giudizi sono favorevoli a questa ultima innovazione, soprattutto quelli dei veri patiti del genere adventure, non si può negare che si è guadagnato molto dal punto di vista del realismo. In risposta a questa vera e propria innovazione, nel settore degli arcade si è assistito a uno spostamento verso un genere di gioco che, paradossalmente, risulta sempre più assomigliante a un adventure.

Il filone classico, quello degli spara e scappa tanto per intenderci, sta cedendo il passo a questa nuova corrente costituita da game, per così dire, più intelligenti. In quest'ottica Hopeless senza dubbio rappresenta quanto di meglio ci si potesse aspettare. E' un arcade che appassionerà senza sconvolgere i più acerrimi difensori degli arcade puri e al tempo stesso i filo-adventure. Per la vastità del campo di gioco e per il tipo di interazione ricorda molto game quali Quakeminus e Delta force, ma in quanto al tema risulta perfettamente originale.

Hopeless inoltre è sicuramente un gioco accattivante e completo sotto ogni punto di vista. La grafica risulta gradevolissima così come gli effetti sonori. Ottima è la giocabilità: l'omino che controllate risponde molto bene ai comandi e sia le armi che la modalità di spostamento sono selezionabili molto facilmente.

Come si gioca

Hopeless è ambientato nello spazio, in una galassia infestata da mostriciattoli di ogni tipo e aspetto. Lo scopo del gioco consiste nel recuperare dei cuori (in realtà sono dei terminali che sul video mostrano un cuore pulsante) sparsi per i vari asteroidi del sistema. Per iniziare la partita basta premere la barra spaziatrice, terminato il caricamento del programma e dato il RUN.

All'inizio del gioco il vostro alter ego, l'omino blu con la tuta spaziale dotata di jet, si trova all'interno di una astronave, peraltro assolutamente inutile dal momento che non si può muovere. La prima operazione da compiere consiste nell'aprire lo sportello dell'astronave, per poter uscire nello spazio, e quindi andare sulla parte posteriore dell'astronave per collegarsi a un secondo terminale. Questo terminale consente di avere una visione di insieme della porzione di spazio in cui ci si trova e quindi di localizzare gli obiettivi della missione.

Per collegarsi al terminale, sia quello che consente l'apertura delle porte dell'astronave sia gli altri sparsi qua e là, bisogna porsi davanti alla tastiera e spostare indietro la leva del joystick premendo contemporaneamente il tasto di fire.

I terminali ai quali ci si può collegare sono di tre tipi. Il primo è costituito da terminali di comando, riconoscibili dalla barra presente sul video, cioè terminali che consentono l'apertura di sbarramenti. Dopo essersi

collegati lo sbarramento controllato verrà aperto o chiuso, a seconda del suo stato al momento del collegamento. Nel caso in cui ci si trovi distanti dallo sbarramento e quindi non si possa vedere se questo è stato aperto o meno basta fare riferimento alla barra presente sul video al termine del collegamento: se è verticale lo sbarramento è stato aperto mentre se è orizzontale significa che è stato chiuso.

Gli altri due tipi di terminali sono quelli di controllo e di teletrasporto. Entrambi sono riconoscibili per le caratteristiche figure presenti sul video: figure geometriche simili a pallini nel primo caso e rombi nel secondo. I terminali di controllo consentono di fissare l'obiettivo che si vuole raggiungere. Una volta collegati viene mostrata la posizione in cui ci si trova, una finestra che mostra una porzione dello spazio circostante e un menù contenente tre opzioni: Home, Set e Quit. La finestra può essere controllata con il joystick e il cursore posto al centro permette di selezionare l'obiettivo prescelto.

La selezione avviene premendo il tasto di funzione F3, l'effetto di questa operazione verrà spiegata in seguito. Si può selezionare un solo obiettivo e una eventuale selezione indesiderata può essere corretta semplicemente premendo nuovamente il tasto di funzione F3 in corrispondenza del nuovo obiettivo. Il tasto di funzione F1 consente di riportare la finestra sul luogo in cui ci si trova. Con il tasto di funzione F5 si ritorna al modo operativo. I terminali di teletrasporto permettono di trasferirsi istantaneamente da un luogo a un altro dello spazio. L'unica restrizione consiste nel fatto che il punto di arrivo deve essere un terminale di controllo (sono indicati da tondini). Anche in questo caso la selezione dell'obiettivo avviene seguendo le modalità viste in precedenza. Per effettuare il teletrasporto basta premere il tasto di funzione F5.

Il pannello di controllo posto sotto la finestra con la situazione di gioco è come al solito importantissimo. Sulla sinistra sono riportate sei icone: quelle più in alto indicano l'arma attiva mentre le tre più in basso indicano lo stato di gioco. Le armi disponibili sono: il campo di forze, il laser e le mosse di karate. Sono selezionabili mediante il tasto di funzione F3 e l'arma attiva viene indicata dall'icona in colore giallo; tenete presente che le tre icone riproducono, partendo da sinistra, il campo di forze, il laser e il karate. Le tre icone rimanenti indicano, partendo da sinistra, il mezzo di movimento attivo, i piedi o il jet, e se si è collegati o meno con un terminale. Solo la modalità di movimento può essere controllata direttamente, premendo il tasto di funzione F1. Anche in questo caso la funzione attiva è indicata dalla icona in colore giallo.

Al centro del quadro di controllo vengono indicati la forza, l'energia e il carburante rimasti. La forza è un parametro che risente unicamente degli scontri con i mostriciattoli e subisce brusche riduzioni quando si entra in contatto con uno di essi. L'energia indica la possibilità di usare le armi e ovviamente è influenzata dall'uso delle stesse. Entrambi i parametri sono vitali nel senso che se giungono a zero si perde una vita. Il carburante è strettamente legato all'uso del jet e quando giunge a zero questo non può più essere utilizzato. Il recupero di un cuore consente di reintegrare automaticamente il livello di questi tre parametri. Tenete presente che per recuperare un cuore basta effettuare il collegamento al terminale che lo mostra sul video. Sulla sinistra del quadro di controllo si trova il pannello che mostra la direzione di movimento.

Il tondino blu posto sul bordo funziona in maniera simile a una bussola: infatti indica la direzione da seguire per raggiungere l'obiettivo selezionato al terminale di controllo. La bussola indicherà sempre la direzione dell'ultimo obiettivo selezionato e una volta raggiuntolo verrà disattivata (quando la bussola è disattivata il tondino blu ruota attorno alle frecce che indicano la direzione).

Punteggio

Ogni mostriciattolo ucciso dà diritto a 100 punti mentre ogni cuore recuperato ne frutta 100.000. Non è previsto nessun bonus di vite in relazione al punteggio ma solo a missione compiuta, cioè dopo aver recuperato tutti i cuori. Il punteggio più alto viene memorizzato e mostrato nel quadro di controllo.

Roberto Meneghetti

Eccoci all'ultima puntata del corso di inglese in alta risoluzione: forma passiva, modo condizionale ...

Computer English

Ecco l'ultima puntata del corso di inglese (le prime due lezioni sono apparse sui numeri di novembre e dicembre mentre la terza puntata è stata pubblicata sul numero di febbraio di quest'anno). Ovviamente, a causa soprattutto del poco tempo che avete avuto a disposizione e del limitato numero di puntate, non ci sentiamo di poter dire di avervi insegnato questa lingua. Lo scopo di questo corso è infatti quello di avvicinarvi a una lingua straniera mediante una didattica innovativa e un apprendimento interattivo che con i mezzi classici di studio risulta alquanto improbabile. Speriamo di essere stati in grado di trasmettervi una conoscenza generale del fonema inglese e di servirvi da stimolo per una crescita in questa direzione.

Caratteristiche generali

Le caratteristiche generali del corso vi sono ormai note. Rivediamole comunque a grandi linee. Si presuppone che chi fruisce di questo metodo abbia una conoscenza di base della lingua inglese; pertanto le nozioni elementari non vengono introdotte. Anche chi non ha mai studiato inglese può però fare buon uso di questo corso, con un pizzico di buona volontà in più.

La durata del corso è prevista in quattro puntate, ognuna delle quali costituita di otto unità (vedi tavola nella pagina seguente); viene proposta una sezione, sempre all'interno di ogni unità, contenente alcune regole grammaticali che saranno introdotte nel dialogo o nella lettura relative all'ambiente stesso.

Traduzione delle unità di lavoro

Per chi ha meno dimestichezza con la lingua pubblichiamo le traduzioni delle otto unità della quarta lezione.

UNIT 1.
Clerk (impiegato): Buon giorno. Posso aiutarvi?
John: Buon giorno. Vorrei andare a Roma.
C.: Come volete andarci, in treno o in aereo?
J: Penso che preferirei andarci in treno.
C.: E' davvero un viaggio piacevole. Sono sicuro che lo gradirete.
J: Suona grandioso. Non ho mai visto l'Italia. Che treni ci sono?
C.: Un rapido parte tutte le mattine alle dieci. Vi è una carrozza ristorante.
J.: Ciò è carino. Quanto costa il biglietto di prima classe?
C.: Cinquanta sterline.
J.: Preferisco prendere un biglietto di sola andata. Può darsi che torni con l'aereo. Grazie. Siete stato molto gentile.
C.: Affatto. Fate buon viaggio.

UNIT 2.
Organizzando qualcosa.
John: Che cosa stai facendo ora?
Peter: Sto tendando di procedere con i miei compiti. Mi sembra di non andare avanti del tutto. Quando studi questo libro non hai mai davvero la sensazione di giungere nel cuore della materia.
J.: Io non mi preoccuperei. Sono sicuro che te la caverai conoscendo appena le cose essenziali quando sosterrai l'esame.
P.: Oh! Il tempo scorre (va avanti), l'esame si avvicina e io non concludo mai nulla.
J.: Bene. Ti darò la risposta. Stiamo organizzando una festa per sabato.

UNIT 3.
L'anno scorso vidi in una libreria un testo molto interessante che avevo sempre voluto leggere. Purtroppo era scritto in cinese e così non potei comperarlo. Due o tre giorni fa John, un mio amico, mi telefonò. Lo aveva visto in uno strano posto, un monastero buddista, ed era tradotto in lingua inglese. John mi disse che questo libro era stato rubato a un interprete. Andai là e lo presi. Adesso è mio, nessuno lo ha più cercato.

UNIT 4.
Avevo appena finito di cenare e avevo appena cominciato a lavare tutti i piatti quando si sentì bussare alla porta. Era mio cugino, che era appena tornato dal Messico. Era ubriaco e mi disse che non sapeva come smaltire la sbornia. Aveva bevuto tequila durante tutte le vacanze. Il giorno successivo doveva andare a lavorare e poiché avevo vissuto la stessa situazione gli insegnai alcuni segreti. Se sieti ubriachi, telefonatemi, vi aiuterò.

UNIT 5.
Mary sta parlando con il suo amico John.
Mary: Vorresti venire da Peter con me?
John: Oh, no. Non voglio. Inoltre è inutile.
M: Perché?
J: Perché John è all'estero. E' partito stamattina.
M: Non ti credo.
J: Come vuoi, ma vedrai che è andato via.
M: E' impossibile. Se fosse partito mi avrebbe chiamata.
J: Va bene. Vai! Avrò finito i miei compiti quanto tu ritornerai.
M: Sei sempre così antipatico con me. Addio.
J: A più tardi, Mary!

UNIT 6.
La notte scorsa feci una passeggiata. Il tempo era tiepido e a me piace molto camminare sotto le stelle. Ma accadde qualcosa di orrendo: persi la strada di ritorno. Avevo camminato per due ore. Cominciai a girare in tondo fino a che non arrivai ad una cabina telefonica. Se avessi avuto una moneta avrei telefonato a qualcuno. Alla fine incontrai un poliziotto e gli chiesi dove fossi. Mi spiegò come tornare a casa. Se non fosse stato distante avrei camminato fino a casa, così presi il treno delle 3 del mattino. Lo aspettai per 3 ore.

UNIT 7.
Quanto lontane, o quanto grandi, sono le cose? Tutti sanno che la maniera per misurare le distanze è di partire con un bastoncino e contare. Cominci con una unità (di misura) e conti. Vi sono cose più piccole o più grandi di altre. La mia casa è più alta della tua, ma il suo cane è migliore del mio. Vi è cosa più interessante di questo? Io non posso immaginare un mondo senza dimensione, e voi?

UNIT 8.
Contenta di vederti.
Mary: Che bello vederti di ritorno a Milano. Dove sei stato durante questo periodo?
John: Sono andato in Africa.
M: Come l'hai trovata?
J: Molto bella.
M: Che cosa stai facendo ora?
J: Sto cercando lavoro, ma è già da un po' di tempo che mi sono annoiato di guardare sul giornale.
M: Mi spiace. Non posso aiutarti.
J: Non importa. A proposito, da quanto tempo abiti in questa parte della città? Ho chiesto a Peter dove tu fossi. Dice di non averti vista da molto tempo.
M: Vivo qui da febbraio.
J: Strano, molto strano. Non ti ho mai visto prima d'ora.

Ogni lezione comprende un vocabolario globale per tutte le unità. Infine una serie di quattro esercizi permette di provare il vostro grado di comprensione delle regole e dei vocaboli di volta in volta introdotti.

E' consigliabile passare all'unità successiva solo quando è ben chiaro quanto esposto nell'unità che si sta analizzando, per non rischiare di accumulare dubbi e incertezze.

Per motivi di ordine tecnico è necessario, quando si risolvono gli esercizi, scrivere in minuscolo tutte le parole (anche quelle di inizio frase) tranne quelle per cui si rende necessaria l'iniziale maiuscola (la prima persona singolare, gli aggettivi di provenienza geografica ecc.). Per quanto riguarda le forme contratte, infine, è bene non farne uso se non nei termini precisati dal corso stesso.

Quarta lezione

In questa lezione parleremo dei tempi verbali non ancora analizzati (il trapassato remoto e il futuro anteriore), della forma passiva dei verbi transitivi, del modo condizionale. Analizzeremo velocemente come si costruiscono i comparativi e i superlativi, senza entrare troppo nel merito delle molteplici eccezioni per evitare una noiosa elencazione di vocaboli.

Come ultimo paragrafo di questo corso abbiamo pensato di segnalare alcuni usi idiomatici del verbo To Get. Il motivo principale di questa scelta dipende dal fatto che, nella lingua parlata, esso è di frequente utilizzo, un po' come da noi il verbo fare.

Ada Bosisio

Sotto sotto funziona così

Si conclude il ciclo sull'anatomia umana con il programma dedicato agli apparati circolatorio, digerente, respiratorio ed escretorio.

L'ultimo programma sull'anatomia umana è dedicato allo studio degli apparati interni.

Come i primi due, rispettivamente sulle ossa e sui muscoli, anche questo programma è strutturato in maniera abbastanza semplice, sia per quanto riguarda la codifica sia per quanto riguarda l'utilizzo.

Il caricamento del programma dalla cassetta allegata è come al solito semplicissimo, anche se questa volta, per necessità grafiche, è stato necessario utilizzare un nuovo set di caratteri. Questo set infatti, trovandosi in RAM a partire dall'indirizzo esadecimale $0800 a $1000 va a occupare i primi 2K dell'area Basic, di conseguenza il codice del programma è stato spostato più in alto a partire da $1001.

Comunque di questo non dovete assolutamente preoccuparvi perché il meccanismo è del tutto trasparente e quindi il caricamento può avvenire con un semplice LOAD, seguito dal RUN, o semplicemente con SHIFT RUN/STOP. Nel nuovo set sono stati sacrificati i caratteri grafici e i reverse per definire quelli necessari alle illustrazioni delle parti anatomiche più complesse e che mal si prestavano a essere descritte con i caratteri grafici standard del C64, come era stato fatto per le puntate precedenti.

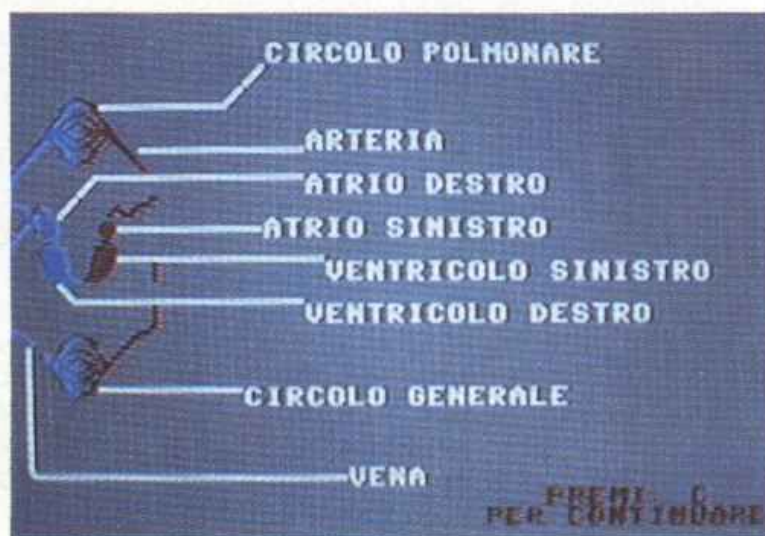
Come funziona

Lanciato il programma apparirà una panoramica della lezione in esame con una visione sommaria, ma completa, dei vari sistemi escluso il circolatorio che, comunque, compare successivamente in dettaglio.

La schermata iniziale si ripeterà finché non premerete il tasto C, come è tra l'altro suggerito da una scritta riportata nella schermata stessa. Dopo essere usciti dalla schermata di presentazione, vi verrà sottoposto un menù principale con due sole possibili scelte, che si commentano da sé: spiegazione o interrogazione. Nel caso scegliate la prima opzione (ve lo consigliamo, se siete alle prime armi), vi verrà mostrato il sottomenù, le cui prime 4 scelte si riferiscono ai vari sistemi del corpo di cui volete prendere visione, mentre la quinta vi riporta al menù principale.

Se avrete deciso di visionare un qualsiasi sistema del vostro corpo (scelte da 1 a 4 nel sottomenù), dovrete attendere che appaiano tutte le nomenclature, dopodiché verrete invitati, non appena pronti, a premere C per tornare nel sottomenù; qui potrete scegliere se visionare un'altro apparato, o tornare al menù principale.

Nel caso scegliate di sottostare ad una interrogazione, una volta selezionata la seconda opzione, verrete invitati a digitare il vostro nome, quindi a premere Return; a questo punto, vi verranno sottoposte 20 domande di diverso genere, al termine delle quali, a seconda del numero di risposte esatte

Spiegazione dell'apparato digerente.

Spiegazione del sistema circolatorio.

L'Africa del Sud della South African Airways

Buona parte del territorio dell'Africa del sud è destinato ai parchi nazionali, conservando così intatte le condizioni ambientali di una flora e di una fauna tra le più ricche del mondo. Alle grandi città di Pretoria, Cape Town e Durban si alternano le famose riserve e i suggestivi paesaggi della Namibia, dello Zimbabwe e del Botswana: un territorio e un ambiente tra i più affascinanti del continente africano: torrenti e cascate, montagne e foreste tropicali, canyon e deserti, colline e praterie. Lo splendore della natura libera, il curioso intreccio di un mondo africano vergine e lussureggiante, l'intensità della vita di tipo americano ed europeo, costituiscono il particolare incanto di questa terra.

L'operatore Gastaldi Tours in collaborazione con la South African Airways vi offre tutte queste meraviglie nella nuova edizione dell'opuscolo "L'Africa del Sud della S.A.A.", rinnovando e ampliando i viaggi già conosciuti e includendo anche stati limitrofi come la Namibia, il Botswana e lo Zimbabwe.

Un'assoluta novità, eccezionale ed esclusiva, per chi desidera vivere una vacanza tutta natura è il "Tour Best Africa" di 22 giorni con accompagnatore dall'Italia che propone una grande avventura per scoprire le bellezze naturali di tutta l'Africa del sud, attraverso 4 Paesi: safari a piedi, in canoa e fuoristrada a diretto contatto con una natura selvaggia e incontaminata, dal Parco Etosha al Parco Kalahari, dal delta dell'Okavango alla riserva di Moremi, dalle cascate Vittoria al fiume Zambesi, senza però tralasciare le stupende città di Johanesburg, Durban e Città del Capo. Le quote per persona in camera doppia partono da L. 5.350.00 in alberghi di 1 categoria o lusso. Chi volesse maggiori dettagli su questa o altre proposte, può rivolgersi presso tutte le Agenzie di viaggio che distribuiscono il programma "L'Africa del Sud della S.A.A." della Gastaldi Tours.

La Ibm Italia premia 25 tesi sull'informatica

Venticinque premi di 6 milioni di lire ciascuno per tesi di laurea su argomenti di informatica saranno assegnati dalla Ibm Italia a giovani che si siano laureati con il massimo dei voti in ingegneria, fisica, matematica, informatica, economia e statistica tra il settembre 86 e il luglio di quest'anno. Obiettivo del concorso, giunto alla terza edizione, è quello di promuovere l'impegno degli studenti nel settore dell'informatica e della telematica. L'iniziativa si inserisce nel quadro delle attività avviate dalla Ibm nel settore universitario e scientifico.

Il telefono diventa un terminale dati

Terminali dati, personal computer, stazioni di lavoro utilizzate alla Italtel possono oggi parlare tra di loro nell'ambito di una rete di comunicazioni integrata voce e dati: il sistema è basato sulla centrale telefonica privata Office BX 10000, realizzata dalla Italtel Telematica, operativa nelle aree di Milano e Castelletto di Settimo Milanese. L'Office BX 10000 consente un servizio dati evoluto nelle funzioni e semplice nell'utilizzo: può trasmettere e ricevere dati, testi e documenti attraverso la rete telefonica interna.

Sotto sotto funziona così

che avrete fornito, vi verrà presentata una valutazione, che potrà essere, a diversi livelli, positiva o negativa.

Commento al listato

Ecco le principali linee del programma, con le relative spiegazioni.

• Linee 1 - 81. Contengono le istruzioni iniziali di routine, l'impostazione delle variabili e la schermata grafica di inizio.

• Linee 96 - 139. Contengono le istruzioni DATA per la schermata grafica iniziale, lette da READ.

• Linee 200 - 250. Contengono le istruzioni per il menù principale.

• Linee 500 - 557. Contengono le istruzioni DATA (seguite da un numero per la localizzazione a video), con le denominazioni dei singoli organi.

• Linee 690 - 870 (divisibili, volendo, in 690 - 770 e 790 - 870). Contengono 2 sottomenù pressoché analoghi, il cui scopo è di sostituire la funzione INPUT con la GET nel dialogo utente-computer.

• Linee 1000 - 1040. Contengono le istruzioni per la visualizzazione del sottomenù e opzioni relative.

• Linee 1097 - 1409. Contengono le quattro subroutine grafiche visualizzanti i diversi sistemi principali, usate sia nella fase di spiegazione sia in quella di interrogazione.

• Linee 1500 - 1530. Contengono quattro subroutine, attribuibili alle rispettive precedenti, la cui funzione è stabilire quali e quante siano le istruzioni DATA delle righe 500 - 557 da visualizzare a seconda delle necessità.

• Linee 1600 - 1640. Contengono le istruzioni per la lettura dei DATA e la loro rappresentazione sul video.

• Linee 2000 - 2060. Istruzioni per la presentazione dell'interrogazione con le opzioni collegate.

• Linee 2110 - 2910. Contengono le istruzioni fondamentali e portanti per la struttura della funzione Interrogazione, e contemplano tutte le possibili eventualità del caso.

• Linee 3000 - 6040. Contengono le istruzioni concernenti la fine dell'interrogazione, di una sua analisi e valutazione finale.

• Linee 4000 - 4040. Contengono subroutine di funzione che si ricollegano alle istruzioni precedenti e successive riguardo la fine dell'interrogazione.

• Linee 5000 - 5030. Contengono istruzioni di funzione che riguardano i rispettivi sistemi di cui sopra.

Chi possiede il drive può trasferire i tre programmi di anatomia e costruirsi un menù principale che lancia la sezione desiderata, con la solita tecnica di linkaggio del Basic.

Fulvio Lupieri

Il mistero di Koko

Un game appassionante, un'avventura all'interno di un labirinto sotterraneo infestato di robot. L'astuzia è d'obbligo per recuperare tutti i tesori...

Il nome, Koko, non lascia intendere nulla di buono. Non promette una tensione mozzafiato e non garantisce affatto emozioni da avventura spaziale. Tuttavia la prima impressione, almeno in questo caso, non è quella che conta. Ci troviamo di fronte a un arcade che sicuramente entusiasmerà gli appassionati dei game intelligenti, in cui i riflessi non sono più importanti di una buona strategia e in cui l'audacia deve lasciare il passo alla prudenza e alla pazienza.

La situazione di gioco è assolutamente originale, ricorda vagamente la Corsa del faraone, e la giocabilità è a un livello più che buono. Non mancano certo le particolarità, messaggi un po' coloriti che sottolineano gli errori commessi durante l'azione, e il livello di gioco è sempre impegnativo; tutti questi elementi, insieme alla grafica decisamente ottima, contribuiscono a rendere Koko molto avvincente.

Come si gioca

Caricato e lanciato il programma viene visualizzata la schermata di presentazione che non dà informazioni su ciò che ci aspetta. Per svelare il mistero di Koko si deve premere il pulsante di fire del joystick, che deve essere posto in porta 2. A questo punto viene visualizzato l'ambiente di gioco, un labirinto sotterraneo.

Lo scopo del gioco consiste nel recuperare una serie di tesori, rappresentati da oggetti preziosi, sparsi qua e là per i cunicoli del labirinto. L'impresa è sicuramente molto ardua sia per il numero dei preziosi da recuperare sia per la vastità del labirinto. Lungo tutto il percorso sono disseminate trappole mortali: fosse piene di lava, barriere elettrificate e robot. Per evitare le fosse si deve compiere un salto, ottenibile ponendo la leva del joystick in posizione diagonale. Prima di effettuare il salto è necessario controllare con cura la propria posizione e, per essere sicuri di raggiungere la sponda opposta, ci si deve mettere sull'orlo della fossa. I robot possono essere evitati solo azionando lo scudo, ogni tentativo di saltarli darà esito negativo e costerà una vita. Per azionare lo scudo si deve abbassare il joystick in diagonale.

Le barriere elettrificate costituiscono l'ostacolo più arduo. Infatti sono visibili solo quando sono attive e non esiste alcuna protezione per superarle indenni. L'unico modo per superare questo sbarramento consiste nel calcolare bene i tempi morti, cioè gli istanti in cui la barriera è inattiva, e quindi passare durante uno di essi.

Oltre agli ostacoli si deve fare estrema attenzione anche al quadro di controllo posto al di sotto della finestra che mostra la situazione di gioco. Contiene tutte le informazioni riguardo l'energia ancora disponibile, le munizioni a disposizione, il tempo rimasto per portare a termine la missione e quello che rimane prima che l'ultima porta aperta si richiuda.

L'energia, indicata da un tratto scuro posto in alto a destra sul pannello, è un fattore molto rilevante del gioco poiché se giunge a zero il vostro alter ego si accascia al suolo segnalando che è stata persa una vita. Il livello energetico scende bruscamente quando si attiva lo scudo e quando, attivato lo scudo, si entra in contatto con un ro-

Caverna di partenza.

bot. E' possibile fare rifornimento di energia raggiungendo delle particolari basi poste in corrispondenza di avvallamenti del terreno. Per reintegrare il livello energetico è sufficiente entrare nell'avvallamento.

Le munizioni sono indicate da un numero posto alla destra della scritta AMMU che individua la quantità di proiettili rimanenti. Da sottolineare che le munizioni servono solo per eliminare i nemici umani che infestano il labirinto e sono completamente inutili contro i robot.

Il rifornimento di munizioni viene fatto automaticamente quando si trova un oggetto prezioso.

Il tempo rimasto per compiere la missione è indicato da 4 cifre consecutive poste accanto alla scritta STATUS. Le due cifre consecutive poste sempre accanto alla scritta Status indicano il tempo che resta prima che l'ultima porta aperta venga richiusa.

Lungo tutto il percorso si possono incontrare due sbarramenti: i muri blu e i cancelli. I primi, riconoscibili dal colore, consentono il passaggio in una sola direzione mentre i secondi, simili a muri di colore giallo, lasciano passare in entrambi i sensi ma solo se se si possiede la chiave.

Una volta aperto, il portone si richiuderà quando sarà trascorso il tempo indicato sul quadro di controllo. Le chiavi per aprire i cancelli sono sparse lungo il percorso e per appropriarsene basta toccarle: in basso a destra verrà visualizzata la scritta lampeggiante KEY.

Punteggio

Un altro importante fattore del gioco è il punteggio, sia perché è legato a una serie di bonus sia perché quello più alto viene memorizzato su disco. Ogni uomo ucciso dà diritto a 20 punti. Ogni chiave recuperata dà diritto a 25 punti mentre il punteggio legato agli oggetti preziosi viene segnalato nel momento stesso in cui l'oggetto viene recuperato.

Conclusioni

Il gioco risulta, per gli appassionati del genere, veramente divertente e completo. Le situazioni di gioco e la grafica stupenda dovrebbero renderlo accettabile anche ai patiti degli arcade del tipo spara e scappa.

Azzeccata l'idea di memorizzare su disco il punteggio più alto. C'è una sola pecca: manca un adeguato accompagnamento musicale in grado di fare da cornice a un quadro di tale valore.

Fabio Galli

Dopo il break out

Chi di noi non ha mai giocato a Break out almeno una volta alzi la mano. Ed ecco ora la continuazione di questo gioco da bar in versione per personal computer.

Questo nuovo videogame, chiamato Arkanoid, può essere definito come la continuazione di Breakout; la Imagine ha infatti convertito questo gioco da bar della Taito per la gioia dei vari possessori di computer che così si potranno divertire in casa loro.

Ed eccoci qui a sbattere freneticamente una pallina contro un muro, spaccando mattone su mattone, sprecando quintali di adrenalina e a volte logorandoci i nervi. La Imagine ha voluto dare a questo gioco un piccolo tocco di spazialità e infatti scoprirete che la barra di comando altro non è che una piccola astronave fuggita dall'astronave madre dopo che questa è stata colpita; e scoprirete anche che la pallina che dovete respingere è un raggio che vi serve per abbattere tutti i mattoni.

Comunque, che si tratti di un gioco spaziale o del figlio ultimo nato di Breakout, resta un gioco appassionante e graficamente ineccepibile; chi ha avuto la possibilità di giocare al coin-op al bar non vi troverà nessuna differenza in questa versione per Commodore 64.

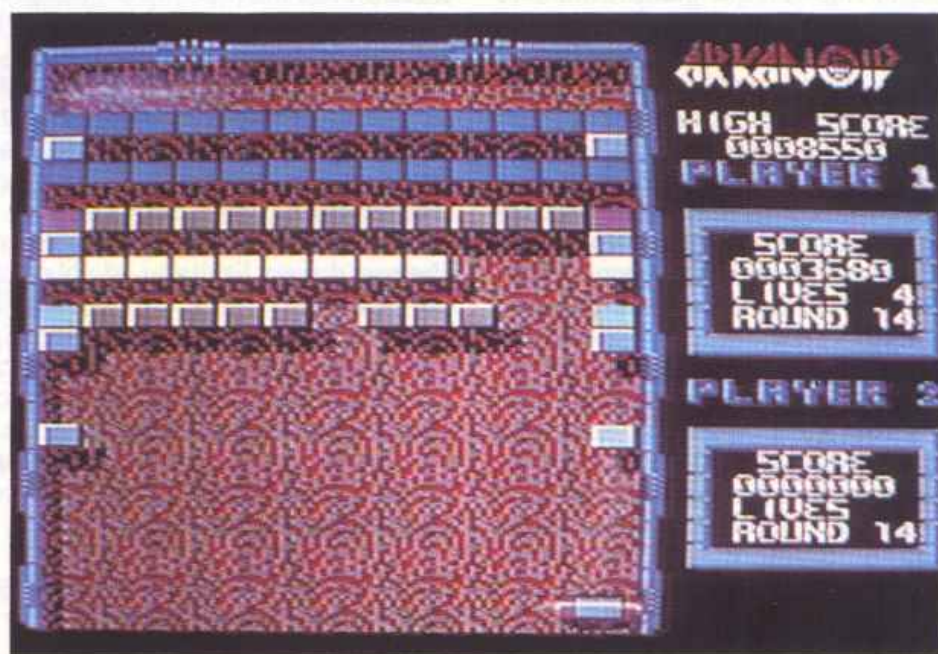
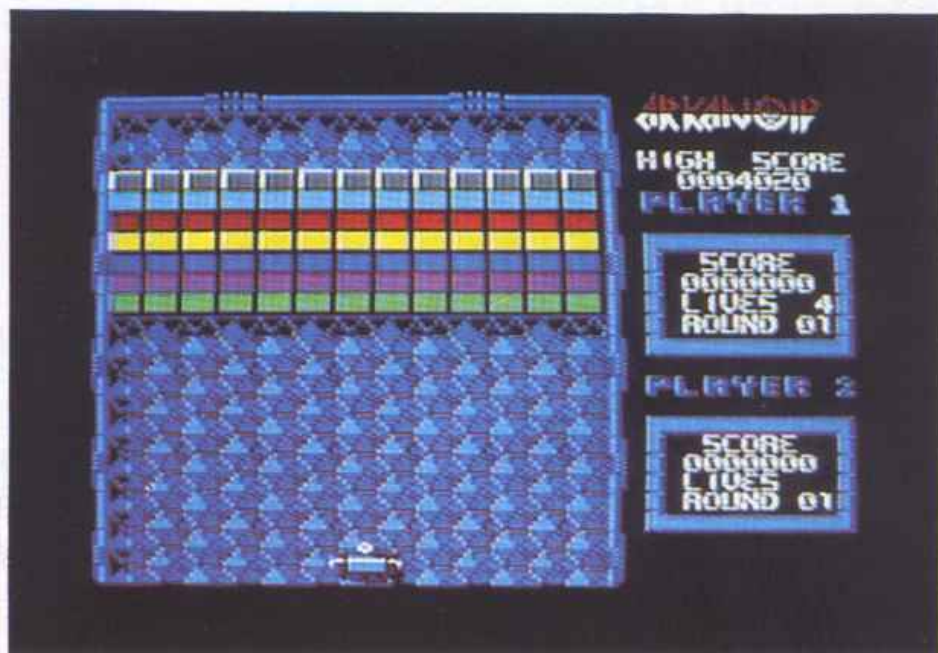
Come si gioca

Siete al comando di una piccola astronave fuggita appena in tempo dall'astronave madre colpita da nemici alieni (anche se l'apparecchio che la colpisce nella presentazione sembra di più un F-15 che non una astronave) e dovete affrontare 32 schermi per poter raggiungere la fase finale.

Finita la carica del gioco vi viene chiesto cosa intendete usare per giocare tra: Neos Mouse, Joystick, Keyboard o Paddles e se i giocatori sono 1 o 2. Fatte queste scelte vi verrà mostrata la scena dell'abbattimento dell'astronave madre e la fuga della piccola astronavicella, che subito si tramuta in racchetta per l'inizio del gioco.

La particolarità di questo gioco è che non dovete limitarvi ad abbattere i classici mattoni, che valgono da 50 a 120 punti a seconda del colore, ma che dovete anche catturare alcune capsule, contrassegnate da varie lettere, che, oltre a regalarvi 1.000 punti, hanno vari poteri e possono cambiare la vostra astronave; le lettere che le contraddistinguono sono:

• **S** - rallenta la velocità del raggio, facilitando il posizionamento.
• **C** - calamita l'astronave in modo che ogni volta che il raggio la colpisce vi rimane attaccata, permettendovi di spostarvi prima di rilasciarlo.
• **E** - ingrandisce l'astronave e permette un minore rischio di non prendere il raggio.
• **D** - divide il raggio in 3, molto utile quando lo stesso è tra molti mattoni, ma assurdo quando il raggio è vicino all'astronave.
• **L** - fornisce l'astronave di un laser in modo da abbattere molti altri mattoni sparando a destra e a sinistra.
• **B** - rompe una sezione di muro permettendo il passaggio al livello superiore molto più velocemente.
• **P** - ogni volta che vedete spuntare questa capsula buttatevi alla sua cattura, perché vi regalerà una vita, ma mettetevi l'anima in pace in quanto è

Arkanoid è distribuito in Italia dalla Mastertronic. E' in vendita a 19.900 lire da Alcor, Corso di Porta Romana 55, 20122 Milano, tel. 02/5450624 e nei migliori computer shop d'Italia.

veramente raro vederla comparire.

Nello schermo vagano poi alcuni moduli alieni, che vi regaleranno 100 punti ogni qualvolta ne colpirete uno con il raggio o con l'astronave, ma osservateli bene perché vi possono giocare brutti scherzi, infatti, se il raggio li colpisce, ritorna indietro molto più velocemente e, se questo succede molto vicino all'astronave, avrete quasi sicuramente perso una vita.

Vi consigliamo l'uso del joystick per questo gioco anche se le prime volte sarà molto difficile prendere le misure esatte per ribattere il raggio, in quanto, una volta compreso bene il meccanismo di funzionamento, il joystick permette una giocabilità maggiore dovendolo muovere esclusivamente a destra e a sinistra e al massimo premendo fuoco quando si cattura la lettera C. Il laser è utilissimo per abbattere i mattoni che normalmente crollano solo dopo molti colpi di raggio. Non tutti però, quindi fissatevi solo su alcuni mattoni in quanto ce ne sono certi che vengono abbattuti quasi subito mentre altri sono indistruttibili.

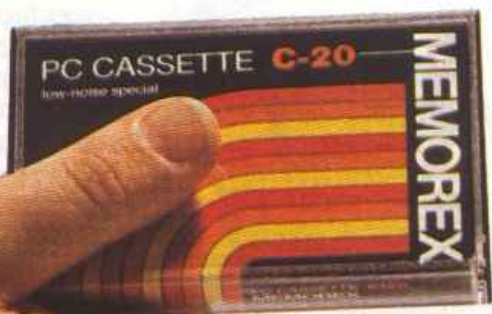
L'unico difetto, se si può chiamarlo così, in questo gioco è, a volte, la velocità del raggio: le prime volte farete un po' di fatica a controllarlo ma poi, una volta presa la mano, sarà molto più facile giocare alle alte velocità che non a quelle basse.

La grafica è ottima e anche la musichetta è buona, insomma il gioco merita la spesa e, se riuscite ad arrivare all'ultimo schermo, potete ritenervi dei veri campioni.

Maurizio Polacco

Un milione per il tuo software

Sì, hai letto bene. Radioelettronica & COMPUTER compensa fino a un milione di lire il software dei lettori. Naturalmente il materiale deve pervenire alla nostra rivista secondo standard ben precisi:

- deve essere corredato del listato su carta;
- deve essere accompagnato dal supporto magnetico (dischetto o cassetta);
- il programma deve essere scritto in modo professionale (niente righe inutili) come pure professionale deve apparire l'impaginazione e la grafica delle videate;
- il tutto deve essere accompagnato da una esauriente spiegazione tecnica di come gira il programma, almeno nelle sue parti principali, e di un articolo che ne spieghi il funzionamento e fornisca le necessarie istruzioni per un giusto impiego.

Fra tutti coloro che invieranno il loro software seguendo queste precise modalità, Radioelettronica & COMPUTER sceglierà quelli da pubblicare. Invierà in tal caso agli autori un regolare contratto che, oltre alle consuete clausole, riporterà l'indicazione del compenso. Detto compenso può arrivare fino a un milione di lire, a seconda del grado di complessità e di professionalità con cui il programma è stato redatto.

Camera oscura

Ingranditori, espositori, luci rosse?
Il controllo elettronico di questi e altri strumenti fotografici è più preciso di quello normale. Realizziamolo con Fotocom.

Fotocom è un apparecchio completamente automatico e indipendente dal punto di vista hardware: se usato, con appropriato software, in abbinamento a un personal computer dotato di porta utente I/O come quella del Commodore 64, è utilizzabile con ottimi risultati soprattutto in campo fotografico, per il controllo di ingranditori, espositori e altri dispositivi come le luci rosse per lo sviluppo delle pellicole o per la stampa di copie.

L'utilizzo a cui il Fotocom è destinato riguarda in particolare modo il campo della fotografia, in quanto un qualsiasi apparecchio a 220 volt collegato all'uscita di pilotaggio può essere agevolmente controllato in stato ON/OFF per un determinato periodo di tempo, anche assai breve (un decimo di secondo). Il collegamento di una qualsiasi periferica, come per esempio ingranditori, sviluppatori o luci rosse, è assai semplice in quanto basta applicare la spina 220 volt alla presa posta sul retro del Fotocom realizzata con due boccole di formato standard. Tutto il controllo dei tempi di attivazione avviene via software, per cui il Fotocom si limita a segnalare, tramite un apposito monitor a tre led, lo stato dell'uscita (ON, acceso, se sono accesi i due led rossi, oppure OFF, spento, se lampeggia il led centrale verde). Ovviamente all'uscita 220 volt può essere collegato un qualsiasi altro tipo di carico che debba essere controllato in accensione/spegnimento per un determinato periodo temporale: l'importante è rispettare la tensione prescritta (220 volt c.a.) e il carico massimo totale sopportabile (5 ampère, l'equivalente di circa 1000 watt).

Il funzionamento si basa sul pilotamento di un'uscita a 220 volt per un periodo di tempo predeterminato variabile da un minimo di un decimo di secondo a un massimo di 59 minuti, 59 secondi e 9 decimi. Particolarmente vantaggiosa risulta essere l'accoppiata hardware-software in quanto ha reso possibile eliminare la sezione circuitale elettronica (altrimenti inevitabile in dispositivi di questo tipo) relativa al controllo delle funzioni: display di conteggio del tempo, selettori e dispositivi di attivazione del clock; il tutto con un layout grafico senz'altro superiore al look dei classici pannelli elettronici di controllo. Problemi di interfacciamento al computer non ne sussistono, in quanto basta inserire l'apparecchio nel connettore della porta utente del computer.

Una sicura interfaccia ottica incorporata, realizzata con un fotoaccoppiatore integrato, garantisce un totale isolamento tra circuito del Fotocom, computer e carico eventualmente collegato all'uscita.

Il circuito funziona direttamente con la tensione di rete 220 volt, grazie all'alimentatore interno di cui è dotato che, oltre a un micro-trasformatore, comprende un regolatore di tensione integrato capace di fornire una tensione continua e stabilizzata a tutto il circuito e allo stadio finale che pilota il carico eventualmente collegato.

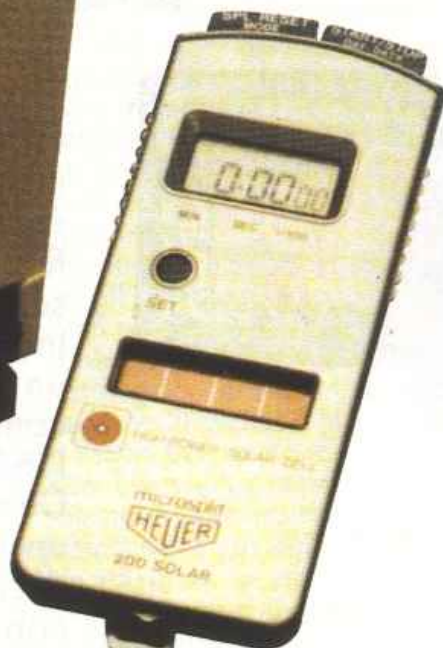
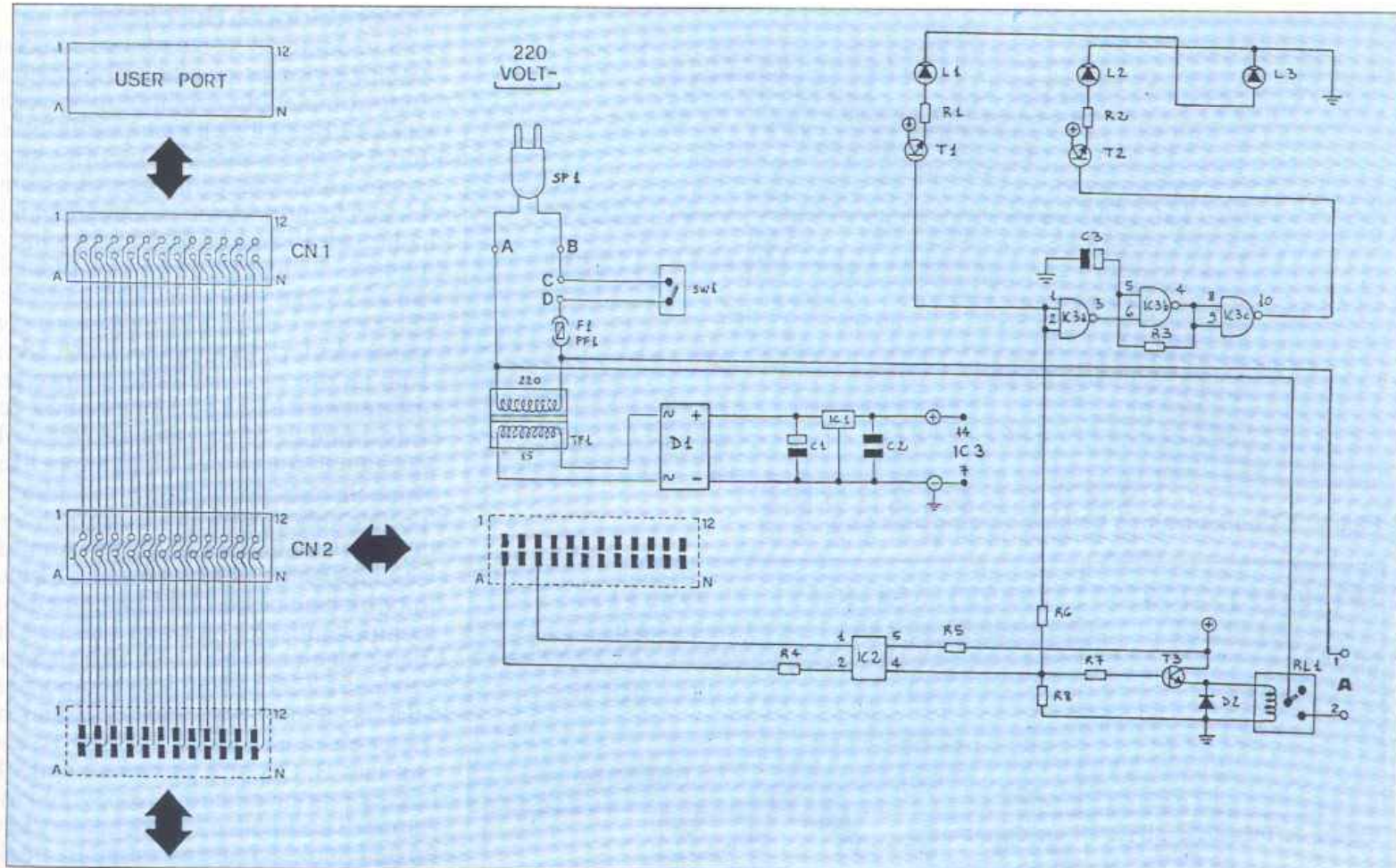



Figura 1. Schema circuitale.

L'apparecchio Fotocom è poi dotato di una line-board duplicatrice che riporta sul retro del contenitore una uscita uguale a quella già presente sul computer: questa caratteristica, molto rara a vedersi su prodotti di questo tipo, permette dunque di collegare in cascata un numero di periferiche anche infinito: chi ad esempio avesse già applicato alla porta utente una periferica, anche non dotata di porta supplementare, potrà toglierla, inserire il Fotocom e quindi ricollegare detta periferica in cascata sul retro del Fotocom. Particolarità molto importante del circuito Fotocom è la possibilità di rimanere connesso anche se non utilizzato: basta infatti disinnescarlo spegnendo l'interruttore principale.

In caso di black out elettrico sulla rete 220 volt nulla accade perché il Fotocom si spegne (come il computer): per ristabilire il controllo dell'uscita di carico basta ricaricare il software. Il funzionamento del Fotocom avviene tramite programmi software più o meno complessi, in base alle prestazioni richieste e alle caratteristiche del computer utilizzato: sono disponibili diverse versioni di programmi, in audiocassette o dischetto, per C64, VIC20 o C128. Un monitor composto da un led verde lampeggiante (power off) permette di segnalare all'utente la presenza della tensione di alimentazione e il corretto funzionamento di tutto il dispositivo: altri due led di colore rosso (power on) segnalano invece, oltre alla presenza dell'alimentazione, anche l'attivazione dell'uscita 220 volt di cui il Fotocom è dotato.

I led di colore rosso e verde funzionano in modo alternato: verde lampeggiante con carico non attivato e rosso per carico attivato.

Schema circuitale: analisi e funzionamento

Il circuito è collegato alla linea dati della porta utente I/O attraverso i due terminali A e C, ovvero massa (A) e PB0 (C): si tratta di una connessione soltanto ottica in quanto è presente un'interfaccia realizzata con il fotoaccoppiatore IC2 che provvede a isolare completamente il circuito del computer da quello del Fotocom.

Quando al pin 1 di IC2 arriva, tramite PB0, un impulso positivo di attivazione, questo viene trasmesso per via ottica al pin 4 del fotoaccoppiatore per poi essere distribuito, tramite R7, direttamente al transistor T3 che controlla il relè RL1 (e successivamente il carico a 220 volt collegato all'uscita A). Lo stesso impulso aziona anche, tramite il transistor T1, i led rossi L1 e L3 collegati in cascata (power on) la cui funzione è quella di segnalare l'innesco del relè e quindi l'attivazione del carico eventualmente collegato all'uscita. Tramite la porta IC3a (parte del CMOS 4093) lo stesso impulso disabilita poi il lampeggiatore NAND IC3b che pilota T2 e quindi il led L2 verde (power off) quando l'uscita non è attivata. Pertanto il funzionamento dei led è alternato: L1 e L2 rossi con uscita ON oppure L2 verde lampeggiante con uscita OFF.

Compito dei tre led è anche quello di segnalare la presenza della regolare tensione di alimentazione. In base allo stato logico presente sulla linea PB0 è dunque possibile attivare o disabilitare l'uscita finale a 220 volt collegata al relè RL1 e visualizzare tramite il monitor a led la situazione che si viene di volta in volta a determinare.

L'interruttore SW1 permette di azionare o disinserire il dispositivo Fotocom.

Il trasformatore TF1 abbassa la tensione di alimentazione in entrata a 220 volt a 15 volt: questa viene raddrizzata dal ponte diodi D1, filtrata da C1 e C2 e stabilizzata dal regolatore IC1 a 12 volt costanti in corrente continua. In serie all'interruttore SW1 è presente anche il fusibile F1 che protegge il Fotocom da eventuali inconvenienti elettrici.

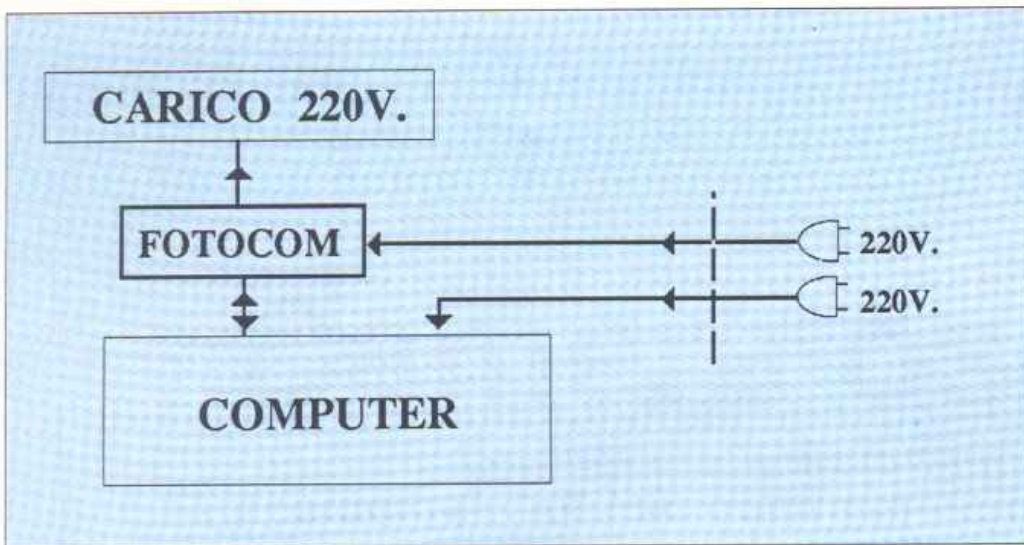



Figura 2. Configurazione tipica di funzionamento.

STATO LOGICO DI PBØ	STATO LOGICO FOTOCOM	USCITA "A"
Ø	Ø	non attivata
1	1	attivata

Figura 3. Tabella analitica di funzionamento.

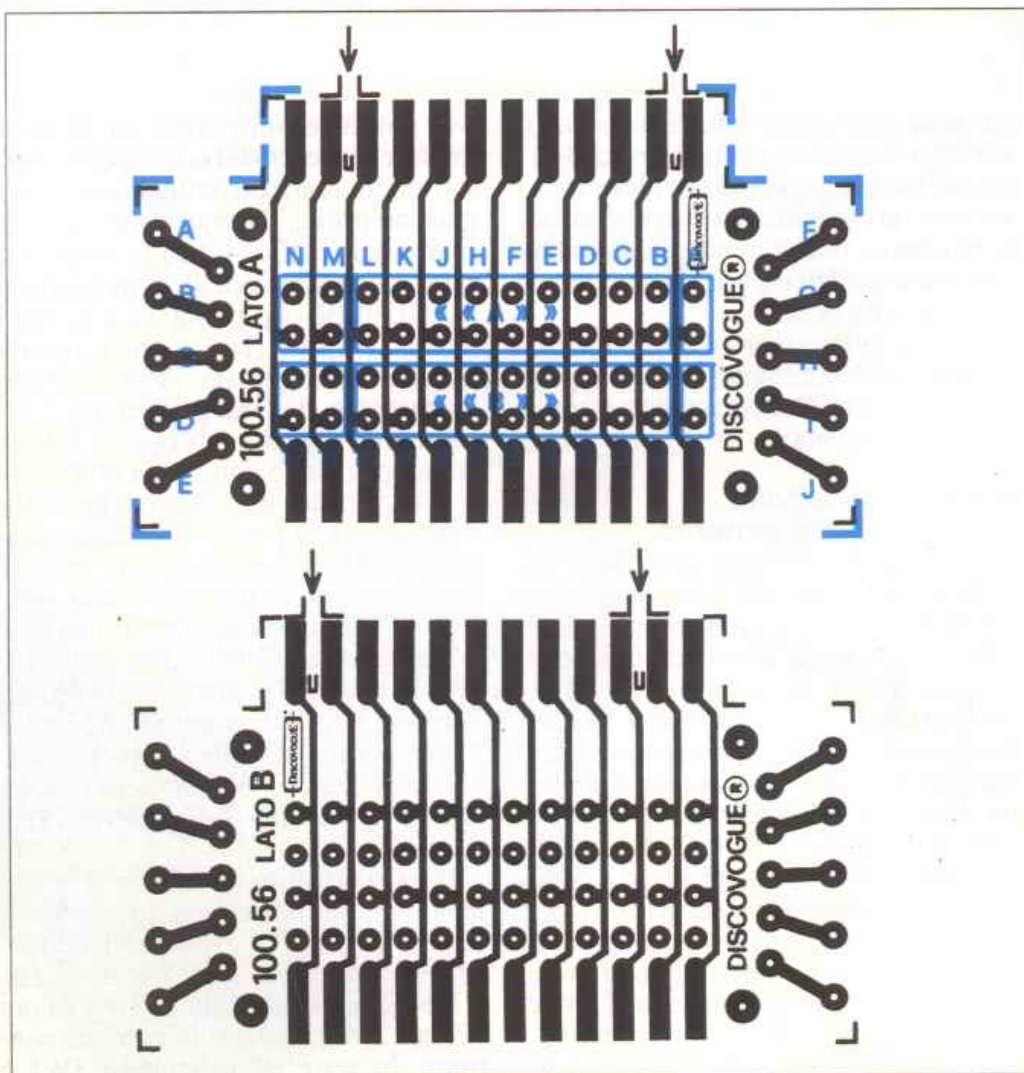



Figura 4. Piste conduttrici dei lati A e B del circuito stampato cod. 100.56 con montaggio dei componenti sul lato A.

Assemblaggio del circuito

Il montaggio del circuito Fotocom va iniziato solo avendo a disposizione tutto il materiale originale, e in particolare i due circuiti stampati bifaccia (codici 100.56 e 137.66), al fine di evitare contrattempi in fase di assemblaggio e indesiderati inconvenienti finali quali il sorgere di disturbi e interferenze durante il funzionamento. Oltre allo stagno e a un saldatore di medio-bassa potenza (30 o 60 watt) con punta preferibilmente piccola, serve solo un piccolo cacciavite a croce per effettuare il fissaggio meccanico del trasformatore al circuito stampato e di questo al fondo del contenitore. Un buon lavoro potrà essere condotto e portato felicemente a termine solo seguendo alla lettera le disposizioni di seguito fornite, senza effettuare alcuna modifica all'ordine di montaggio della componentistica: si tratta infatti, e in particolar modo per il circuito stampato cod. 137.66, di saldare componenti che in alcune zone sono presenti in notevole quantità sia sul lato A che sul lato B. Occorre pertanto evitare di causare accavallamenti e sovrapposizioni di componenti, nonché di creare zone circuitali rese inaccessibili al saldatore dal fissaggio di componenti come TF1,PF1, oppure il relè RL1 e gli integrati. Si raccomandano inoltre saldature veloci, sicure e con poco stagno, da eseguire senza mai persistere su uno stesso punto per più di 4 o 5 secondi, in particolar modo per quanto riguarda i pin degli integrati e i terminali del diodo D2 o dei led. Iniziare montando i due connettori al circuito stampato portante (cod.100.56): CN1 va fissato orizzontalmente, saldando la prima fila di 12 terminali al lato A e la fila inferiore degli altri 12 al lato B; CN2 va invece montato assialmente, infilando i 24 terminali dal lato A in modo che sporgano dal lato B, scegliendo la prima doppia fila di fori (quella anteriore contrassegnata con B) delle due disponibili sul lato A (le serigrafie le indicano chiaramente con due riquadri A e B). Mentre il connettore CN1, a montaggio ultimato, risulta essere comunicante con l'esterno e va applicato alla porta utente del computer, CN2 rimane invece all'interno del contenitore, collegato al circuito stampato principale (cod. 137.66) a esso applicato. A questo punto si può passare al montaggio dell'altro circuito stampato, che raccoglie la maggioranza di componentistica e dunque comprende la parte di circuito hardware funzionalmente più importante. Vanno innanzitutto montati i 6 chiodini terminali capicorda, infilandone 2 dal lato A (punti C e D) e 4 dal

lato B (punti A, B, A1 e A2), in modo che sporgano dai lati opposti, dove vanno effettuate le saldature fissanti. Si può quindi procedere al montaggio del trasformatore TF1 che andando sul lato B (con saldature sul lato A) non costituisce un ingombro ma serve anzi da piedistallo rotante al circuito stampato stesso, agevolando tutte le successive operazioni di saldatura in quanto il circuito stampato potrà essere appoggiato rimanendo sollevato anche senza essere tenuto tra le mani o su un supporto.

TF1 va anche meccanicamente fissato con due piccole viti da inserire negli appositi fori laterali previsti (inserimento e fissaggio con cacciavite vanno effettuati sul lato A in modo che le viti fuoriescano dal lato B). Segue il montaggio di tutti i componenti da sistemare sul lato B, con saldature da effettuare sul lato opposto A. Nell'ordine: ponte diodi D1; condensatore C1 (con il positivo verso D1); regolatore di tensione IC1 (con l'aletta metallica verso C1); relè RL1 e condensatore C2; portafusibile PF1.

Si prosegue quindi montando la rimanente componentistica sul lato A, con saldature da effettuare sul lato opposto B (tranne che per il diodo D2 da saldare sullo stesso lato A. Nell'ordine: circuiti integrati IC2 e IC3; resistenze R4, R5 e R3; condensatore C3 (con il positivo verso IC3); resistenze R8, R7 e R6; transistor T3, T2 e T1; resistenze R1 e R2; diodo D2 (terminale positivo verso L2).

Il montaggio va ultimato con i diodi led L1, L2 e L3 da fissare sempre sul lato A (saldature sul lato opposto B) con i terminali piegati a 90° in modo che le incapsulature di colore rosso e verde possano sporgere all'esterno del circuito orizzontalmente e rimangano sollevate di qualche millimetro, così da fuoriuscire poi dal pannello serigrafato del coperchio del contenitore (attraverso gli appositi fori). Come indicano le serigrafie del circuito stampato cod. 137.66 per tutti i componenti (a eccezione del diodo D2) è previsto il montaggio assiale perpendicolare al piano circuitale: questo sistema, unito alla doppia utilizzazione dello stampato (i componenti occupano entrambi i lati A e B) ha permesso di ridurre al minimo l'ingombro volumetrico dell'apparecchio Fotocom. A questo punto si collega ai due terminali C e D del lato A l'interruttore SW1, tramite un trancio di piattina bipolare di qualche centimetro. Si collegano infine il cavo di alimentazione SP1 ai terminali A e B (lato B) e le 2 boccole alla coppia di terminali A1 e A2 presenti sul lato B, tramite un trancio di cavetto bipolare. Ricordate di inserire il fusibile F1 tra le alette-fermaglio di PF1. In fase di sistemazione definitiva l'interruttore SW1 sarà fatto fuoriuscire dal primo foro anteriore dei cinque presenti sulla sinistra del contenitore. Le due boccole vanno invece montate negli appositi fori presenti sul retro del contenitore: essendo la loro distanza quella standard delle prese-spine 220 volt, sarà subito possibile l'allacciamento con la periferica da pilotare tramite normali spine 220 volt.

Figura 5. Pista conduttrice del lato A del circuito stampato cod. 137.66 con montaggio dei componenti sul lato A.

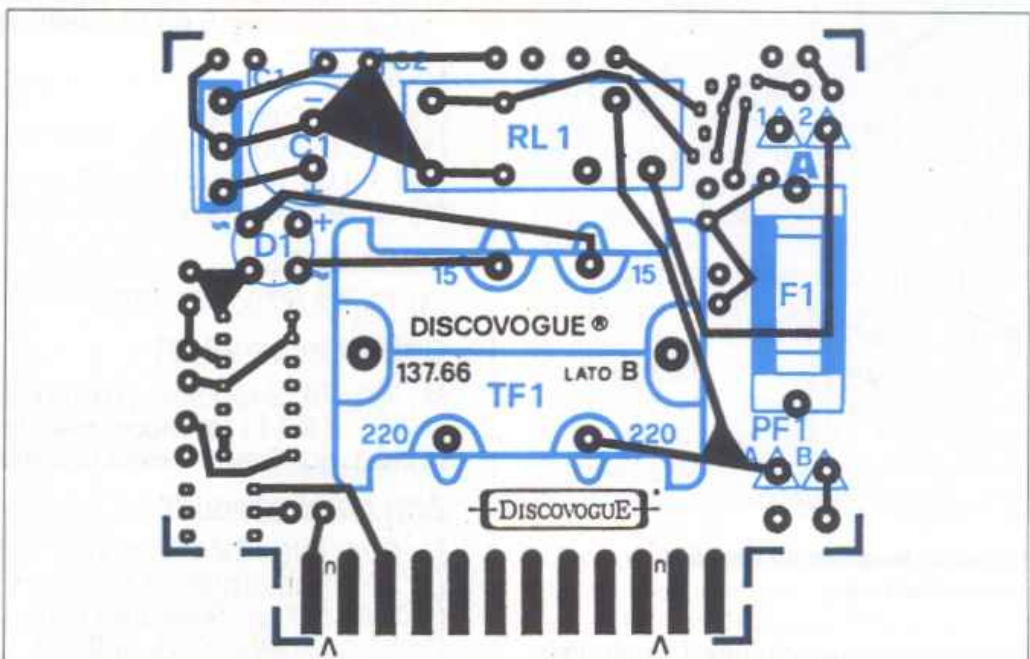



Figura 6. Pista conduttrice del lato B del circuito stampato cod. 137.66 con montaggio dei componenti sul lato B.

Verifica, collegamenti e collaudo

A montaggio ultimato è consigliabile eseguire, tramite un tester, una semplice verifica di corretto funzionamento del Fotocom e un collaudo finale, prima che l'apparecchio venga definitivamente inserito e fissato nel contenitore e quindi connesso al mondo esterno (rete 220 volt, computer a carico di pilotaggio). Con il tester predisposto sulla corrente alternata si darà alimentazione collegando alla rete 220 volt la spina del cavetto SP1 connesso ai punti A e B del Fotocom, verificando poi la presenza della tensione sia sui detti punti A e B del circuito stampato cod.137.66 (lato B) sia sui capi dell'avvolgimento primario di TF1 (controllare che l'interruttore SW1 sia acceso); analogamente sul secondario di TF1 sarà presente una

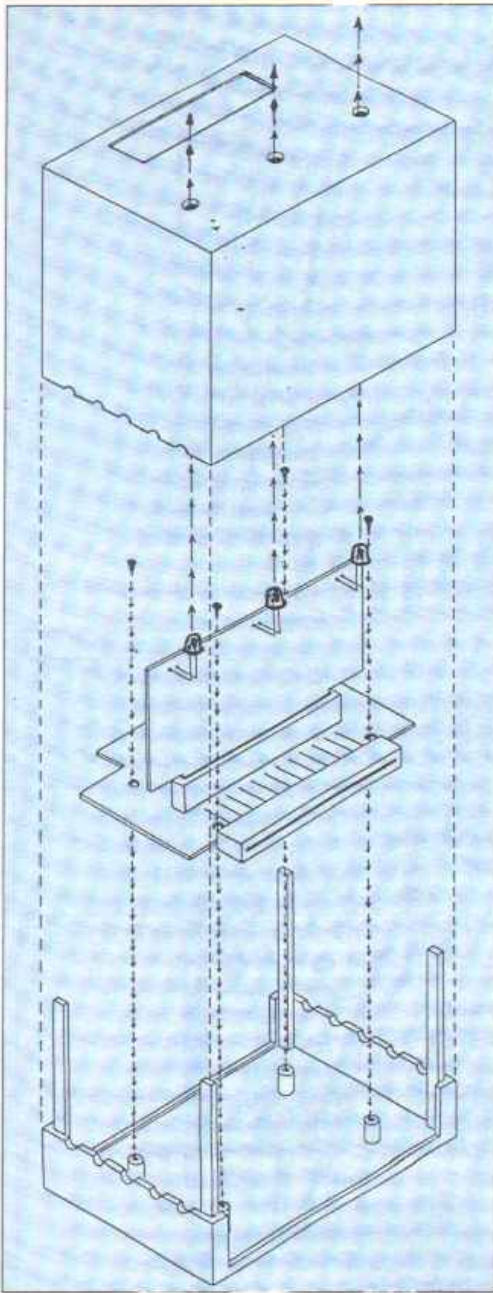



Figura 7. Inserimento del circuito nel contenitore.

Elenco dei componenti hardware

Il circuito è composto in prevalenza da chips integrati, per cui i componenti esterni come resistenze e condensatori risultano essere di numero limitato. Molto sviluppata è la parte elettro-meccanica comprendente 2 circuiti stampati, 2 connettori, il contenitore plastico ABS forato e serigrafato e prese per l'uscita a 220 volt posta sul retro. L'elenco componenti di seguito riportato indica rispettivamente: quantità per ciascun tipo e valore di articoli (numeri tra parentesi); codici circuitali; valori espressi nell'unità standard. I limiti massimi di tolleranza si intendono 5% per le resistenze e 10% per i condensatori.

Resistenze (8)

(1) R1: 390 ohm 1/2 W
(1) R2: 560 ohm 1/2 W
(1) R3: 270 Kohm 1/4 W
(3) R4, R6 e R7: 1 Kohm 1/4 W
(1) R5: 10 Kohm 1/4 W
(1) R8: 100 Kohm 1/4 W

Condensatori (3)

(1) C1: 470 microF 35 VL elettr. vert.
(1) C2: 100 nanoF 100 VL poliest.
(1) C3: 2,2 microF 63 VL elettr.vert.

Diodi (5)

(1) D1:W02 ponte
(1) D2: 1N4004
(2) L1 e L3: led rotondo Ø 5 mm. colore rosso
(1) L2: led rotondo Ø 5 mm. colore verde

Transistor (3)

(3) T1, T2 e T3: BC547B

Circuiti integrati (3)

(1) IC1: 7812 regolatore di tensione
(1) IC2: TIL111 fotoaccoppiatore
(1) IC3: 4093 quad NAND Schmitt trigger

Altri componenti (8)

(1) RL1: relè 12V 1 scambio
(1) TF1: trasformatore miniatura 220/15 VL 160 mA
(1) SW1: interruttore unipolare miniatura
(1) F1: fusibile 250 VL 500 mA
(1) PF1: portafusibile c.s.
(1) SP1: cordone di alimentazione 220 VL con spina
(2) CN1 e CN2: connettore 12+12 poli passo 3,96 mm.

Vari (17)

(1) circuito stampato cod.100.56
(1) circuito stampato cod. 137.66
(4) viti di fissaggio per c.s.
(6) chiodini terminali capicorda c.s.
(2) boccole isolate 220 volt
(1) trancio cavetto bipolare 220 volt cm. 10
(1) trancio piattina bipolare cm.10
(1) contenitore ABS forato e serigrafato cod.137.21

tensione alternata di circa 15 volt. Predisponendo poi il tester sulla corrente continua si verificherà la presenza di una tensione di circa 21 volt ai capi di C1, e di 12 volt a valle del regolatore di tensione integrato IC1. Lo stesso valore (12 volt) dev'essere presente anche sui collettori di T1, T2 e T3 e sul pin 14 di IC3. Occorre quindi constatare il corretto funzionamento del clock costruito su IC3b, rilevando sul pin 4 di IC3 un segnale di clock di circa 4 Hz quando è presente l'impulso positivo sul pin 6 di IC3 (ingresso abilitazione clock). Se tutte le misure danno esito positivo il circuito può a questo punto essere inserito nell'apposito contenitore, fissando il circuito stampato portante (cod.100.56) al fondo tramite 4 viti da inserire negli appositi pilastrini. Il contenitore può essere chiuso accostando il coperchio al fondo, facendolo scorrere sulle quattro sporgenze-guida di plastica, in modo che a fine corsa i tre led sporgano correttamente dai relativi fori presenti sul pannello frontale superiore serigrafato, e che l'interruttore SW1 e il cavetto di alimentazione SP1 possano fuoriuscire dai rispettivi fori presenti sul lato sinistro del contenitore stesso. Anche le due boccole 220 volt del canale di uscita dovranno regolarmente sporgere sul retro. Per un fissaggio definitivo e a prova di manomissione basta applicare in fase di chiusura alcune gocce di collante rapido sulle quat-

tro guide di plastica. Successivamente si applica Fotocom alla porta utente del computer (tramite inserimento del connettore CN1 che, anche a scatola chiusa, fuoriesce in parte dalla relativa finestra anteriore): questa operazione va ovviamente eseguita con apparecchi spenti.

Il collaudo definitivo avviene collegando una periferica a 220 volt all'uscita del Fotocom (facendo particolare attenzione in quanto il circuito è sotto tensione): accendendo il computer e caricando il software dovrà essere possibile pilotare il relè di connessione tra il sistema computer-Fotocom e il carico. Durante il funzionamento il led verde lampeggiante e i led rossi segnalano la regolare presenza della tensione di alimentazione, oltre che la condizione determinata (ON/OFF).

Software dimostrativo

Nella cassetta allegata alla rivista trovate un programma semplice ed efficace che, pur essendo costituito da normali routine, riesce a gestire in modo completo e autonomo la linea dati della porta utente collegata al Fotocom, permettendo in pratica l'elaborazione automatica del segnale di commutazione presente in output e su PB0, e quindi il controllo dell'uscita del Fotocom a cui sono collegati carichi a 220 volt che necessitano di attivazione temporale predeterminata con precisione al decimo di secondo, come apparecchi fotografici quali ingranditori, espositori, luci ambientali e sviluppatori. Il programma occupa un limitato spazio di memoria e si carica in breve tempo con i consueti comandi di Loading previsti dal sistema operativo Basic del computer utilizzato. Non appena si dà il Run appare sullo schermo la videata di controllo, con il data-set di copyright, il display del tempo di riferimento e il menù di opzione, quest'ultimo comprendente le seguenti tre voci operative:

<E> - Spostamento cursore
0/9 - Input valori
SPACE - Conferma e start

La scelta del tempo di attivazione del carico collegato al Fotocom avviene modificando i valori del tempo di riferimento, inizialmente azzerati (il display indica 00:00:0, nell'ordine decine di minuti, minuti, decine di secondi, secondi e decimi di secondo): per l'input è sufficiente spostare il cursore (contrassegnato con la freccia in su) tramite i tasti ≤ (sinistra) e ≥ (destra) e specificare il valore attribuito premendo un tasto numerico compreso tra 0 e 9 (tra 0 e 5 se si tratta di decine di minuti o decine di secondi). La conferma va data premendo Space (la barra spaziatrice), che attiva automaticamente lo start (cioè l'inizio del conteggio) e quindi l'attivazione dell'uscita del Fotocom: sul video scompare il cursore di selezione (freccia in su). Il programma è ovviamente già predisposto per non considerare tempi nulli (00:00:0): l'attivazione è possibile da un minimo di un decimo di secondo (00:00:1) al massimo previsto di un'ora meno un decimo (59:59:9), il tutto con la massima precisione operativa e tolleranze del tutto trascurabili, sia sui brevi sia sui lunghi periodi. Quando termina il conteggio del clock il cursore ricompare sul video ed è possibile ripetere un'attivazione con lo stesso tempo semplicemente ripremendo Space, oppure effettuare variazioni nel modo già visto tramite i tasti ≤, ≥ e quelli da 0 a 9.

La possibilità di effettuare ripetizioni di attivazione con gli stessi tempi senza dover reimpostare ogni volta i parametri è molto utile in campo fotografico, ad esempio per lo sviluppo delle stampe di una stessa foto. Di pari efficacia risulta essere la possibilità offerta dal programma di correggere il tempo semplicemente variando i singoli valori: ad esempio per passare da 2 minuti e 59 secondi (02:59:0) a 2 minuti e 49 secondi (02:49:0) basta modificare soltanto il terzo valore (le decine di secondi) portando il cursore sotto al 5 e sostituire con 4.

Caratteristica è la videata del programma, che rimane costantemente a fondo nero e scritte rosse, in modo da essere al limite pienamente utilizzabile dentro la stessa camera oscura se si opera in fotografia. Anche il Fotocom collegato al computer segnala l'attivazione dell'uscita a 220 volt tramite due led rossi (on), mentre in caso di spegnimento è attivato il led verde lampeggiate (off).

Daniele Malavasi

SE HAI PERSO UN NUMERO...
... HAI PERSO UN TESORO

Ti sei perso un numero, o addirittura più numeri, di RadioElettronica&COMPUTER? In queste pagine ti viene offerta l'opportunità di rimetterti in pari. Di ogni arretrato troverai l'elenco dei progetti pubblicati quel mese. Affrettati a spedire la richiesta utilizzando il buono pubblicato nella pagina seguente: riceverai subito a casa tua il numero o i numeri che ti interessano senza aggravio di spese postali.

Gennaio 1985 - L. 5.000 - Per Commodore 64, Spectrum e Vic 20: un data base per archiviare dischi, libri e cassette. 124 Led e un Commodore in sinfonia. Reset per Commodore con un jack e un pulsantino. Spectrum 48K: la superquaterna. Una routine per avere sullo schermo tutte le memorie Rom e Ram. Programmi per Commodore 64 e Spectrum: un calendario elettronico. Per Vic 20: decodificare le resistenze partendo dal valore o dai colori. Rally automobilistico. Programmi per Commodore 64: un grande artista. Trasformare un vecchio registratore in una segreteria telefonica. IDEABASE: quattro facili progetti per annullare ogni distanza. Le guide di Radio Elettronica &Computer: come ricevere dai satelliti artificiali.

Febbraio 1985 - L. 5.000 - Se la cassetta non carica bene, la colpa può anche essere del registratore, anzi di una semplice vite... Per Commodore 64 e Vic 20: come far partire la lavatrice o bagnare i fiori anche se si è fuori casa. Videogioco per Spectrum: siete dei buoni fantini e scudieri? Un programma di calcolo combinatorio per fare 13 al totocalcio. Un eclettico printer/plotter per C64 e Vic 20. Programmi per Vic 20: come leggere presto e bene. Videogioco esotico tra fiumi e coccodrilli. Per Commodore 64: un generatore di onde quadre ad alta precisione. Panorama sugli MSX. Le guide di Radio Elettronica &C: i fototrasduttori.

Marzo 1985 - L. 5.000 - Per Commodore 64, Vic 20, Spectrum e Apple. In regalo le adesive per cassetta. Per tutte le bici un tachimetro che è quasi un computer. Commodore 64: la vostra orchestra sinfonica. Per gestire la vostra biblioteca un data base che non dimentica nulla. Vic 20: effetti sonori speciali. Uscire dal labirinto evitando trappole e trabocchetti. Spectrum: la versione riveduta e corretta del famoso videogioco del muro, della palla e del rimbalzo. Un data base per mettere ordine nel magazzino di componenti elettronici. Apple: un esperimento di connettore I/O per rilevare il valore di input delle paddle. Versione computerizzata del famoso tris. Tutti i segreti del printer/plotter 1520.

Aprile 1985 - L. 5.000 - In regalo un praticissimo porta-cassette. Disequazioni per chi studia, Slot machine per chi non studia e Grafica per chi disegna: questi i programmi per Vic 20. Per Spectrum ci sono Attacco atomico e Insiemistica, oltre ai segreti per interfacciare da sé. Diesel o benzina?, Omino mangiacrani, Gestione magazzino e soprattutto l'eccezionale Budget familiare per Commodore 64. Come resettarlo? Non manca l'appuntamento con le meraviglie del printer/plotter 1520.

Maggio 1985 - L. 5.000 - Parliamo subito di giochi: Eremon e Drive Quiz per Spectrum, Poker per Vic 20: fanta-

stici. E poi Geografia e Basic italiano per Commodore ed Etichettacassette per Spectrum. Due servizi sul 1520 e sul modem. E una novità destinata a durare: lo standard MSX dalla a alla zeta, ogni numero una puntata. Continua l'omaggio del portacassette per tutti i lettori.

Giugno 1985 - L. 5.000 - Ancora in regalo il portacassette componibile. Magic desk mette a disposizione del Vic 20 le ottime prestazioni del software Commodore. Al Commodore 64, in compenso, un bel regalo: tutto sui cocktail, come prepararli e quando berli. Per Spectrum, Grafica e Ramino. Black Jack per C 16, Antifurto software per C 64, le memorie per Vic 20 in offerta speciale, una succosa puntata sul Basic per MSX e una guida di Radio Elettronica &Computer alla comprensione di una scoperta rivoluzionaria: le fibre ottiche.

Luglio/Agosto 1985 - L. 5.000 - In linea perfetta con la dieta per C 64: personalizzata e rigorosa, ma senza troppi sacrifici. Per Spectrum una carrellata di rebus, per giochi senza fine, e un divertente test da fare con gli amici: Sei creativo? Utility bellissime per Commodore 64 (L'assembler) e Vic 20 (Fast Loader, per abbattere i tempi di caricamento). E ancora: due nuovi programmi per il printer/plotter 1520, un'interfaccia per C 64 e uno splendido programma di grafica per Vic 20. Continua il corso di programmazione per lo standard MSX. Continua anche l'omaggio del portacassette.

Settembre 1985 - L. 7.000 - In questo numero l'eccezionale iniziativa della cassetta con tutti i programmi (ma chi vuole far da sé trova sempre le istruzioni relative ai listati!) per Commodore 64 (Gestione automobile, Totocalcio e Grafica) e Spectrum (Mastermind, Costi postali e Totocalcio). Per Vic 20 un'idea divertente: fai da te le previsioni del tempo. Il corso sul Basic MSX è ormai giunto ad affrontare i programmi più evoluti, con subroutine, loop e scelte logiche. Due giochi nuovi per il plotter 1520 e un'idea per il vostro hardware: un ricaricatore per pile.

Ottobre 1985 - L. 7.000 - Per lo Spectrum, ecco la prima puntata di un corso di inglese, un programma super rapido per risolvere i calcoli geometrici e un'agenda intelligente, simpatica da usare. Per Commodore 64 una fantastica batteria, un programma che sostituisce il registro a scuola e un gestionale di prim'ordine: le spese condominiali per il riscaldamento; per il Vic un magazzino agile e molto funzionale e un bellissimo programma di grafica. Infine la presentazione dell'ultimo Atari e ancora nuovi comandi Basic MSX.

Novembre 1985 - L. 7.000 - Continua per Spectrum il corso di inglese in tre puntate; per la versione 48 K c'è un word processor completo e per l'inespanso un gioco luminoso per parolare in allegria. Per C64: un potente sistema per Enalotto, una utility per creare effetti grafici e sonori senza Simon's Basic, un gioco per insegnare l'aritmetica ai bambini e un altro per divertirsi con le tre carte; e inizia il corso di Pascal! Organo e bilancio familiare con Vic 20, Basic con l'MSX e, per chi ha acquistato il numero di settembre, un aggiornamento del programma Postaspeed.

Dicembre 1985 - L. 7.000 - Oltre all'ultima puntata del corso di inglese, per Spectrum un nuovo programma per la gestione dei campionati di calcio e un gioco/quiz per viaggiare mettendo alla prova le nozioni di geografia. Commodore 64: come ottenere l'effetto ombra; un programma per calcolare gli interessi e l'ammortamento di un debito; un gioco di destrezza, da fare in due. Per Vic 20 una superagenda telefonica e un gioco che richiede abilità, tattica e pazienza: le torri di Hanoi. Continuano gli articoli sul Pascal per C 64 e sul Basic MSX.

Gennaio 1986 - L. 9.000 - Per C64 un ottimo word processor; un rinumeratore intelligente; il gioco della briscola in versione rimodernata; la terza puntata del linguaggio Pascal e un oroscopo personalizzato e scientifico. La tastiera di un sintetizzatore musicale per C16. Per Vic 20 una utility per controllare il perfetto funzionamento del joystick e il gioco La coda del serpente. Infine per Spectrum: una pagella scolastica elettronica e una corsa a ostacoli in alta risoluzione.

Febbraio 1986 - L. 9.000 - La gestione della grafica e del colore con l'MSX. Un gioco strategico: il giro d'Italia in una stupenda realizzazione grafica; il tuo joystick diventa un mouse; quarta puntata del Pascal per C64. Ecco Trivia, il favoloso gioco di grande successo, anche per C16. Per Spectrum un superextragame: 9 giochi in uno! E per finire una occhiata alle caratteristiche software e hardware del Sinclair QL.

Marzo 1986 - L. 9.000 - Per C64 un ottimo programma per gestire un magazzino o una biblioteca. Bellissimo il gioco che insegna l'alfabeto ai più piccini. Finalmente più ordine nei dischetti con un'utility per la gestione del drive. Seconda puntata sulla gestione della grafica e del colore con l'MSX. Per Spectrum: un cruciverba elettronico e un programma per gli appassionati di astronomia; inoltre tutta la gamma delle elaborazioni grafiche messa a vostra disposizione. Versione per C16 del programma budget familiare. Per finire la seconda puntata sul computer QL.

Aprile 1986 - L. 9.000 - Grafici tridimensionali per Commodore 64. Un programmino dedicato all'uso delle spezie in cucina. I quiz per l'esame teorico della patente di guida. Come stampare i vostri disegni creati in alta risoluzione. E inoltre: inizia da questo numero un corso di programmazione in linguaggio macchina. Tutti i colori del sistema MSX nella gestione grafica. Spectrum: gestione dei testi con questa nuova utility. Rischiatutto in edizione da computer. Seconda parte del cruciverba elettronico. Psicotest sulla sensualità per Vic 20. QL: la caratterizzazione del video.

Maggio 1986 - L. 9.000 - C 64: Formattazione a tempo record dei dischetti; un bruco affamato di mele e circondato da un recinto elettrificato; slalom sugli sci nel bosco; altri quiz per l'esame teorico della patente; un editor per sprite e caratteri; seconda puntata del corso di programmazione in linguaggio macchina con l'organizzazione interna del 6510. La gestione del suono nei sistemi MSX. Spectrum: Automobile, quanto costa assicurarla; ecco la terza parte del cruciverba elettronico; un divertente programma di geografia per conoscere l'Europa.

Giugno 1986 - L. 9.000 - Tre utility in linguaggio macchina: StopList, StopSystem e On Error Goto; ottimizzate con la pianificazione computerizzata la vostra serata davanti alla TV; è arrivato Game Maker, il favoloso programma per la creazione professionale di giochi; Guerra Napoleonica (gioco strategico per due persone) e Labirinth (il misterioso labirinto vissuto dall'interno) sono due giochi per le vostre serate; un'utility per la gestione dei file ad accesso casuale; terza puntata del corso di programmazione in linguaggio macchina per Commodore 64. Ultima puntata sullo standard MSX con l'istruzione Sound. Spectrum: gestione di un intero campionato di Formula 1; impariamo il judo con l'aiuto del computer. Grafica di solidi tridimensionali per Spectrum 16.

Luglio/Agosto 1986 - L. 9.000 - Rinnovata e coloratissima versione di Master Mind per Commodore 64. E inoltre: Check Error, utility per il controllo del drive che fa capricci; Osare per vincere, un nuovo adventure a colori; Guerre Stellari: alla conquista di un asteroide; International Karate per gli appassionati di arti marziali; come farsi in casa un ottimo gelato. Quarta puntata del corso di programmazione che completa la presentazione delle istruzioni del set. Geos: il nuovo sistema operativo del C64 uti-

lizza icone e menu pull down. Pagdat è indispensabile per chi deve inserire o modificare dati su una maschera di input a tutto schermo sullo Spectrum 48; chiedi consiglio al computer per scegliere dove passare le tue vacanze; secondo round per gli aspiranti campioni di judo. Gestire gli input da tastiera e i dati in arrivo dalle periferiche con i comandi funzione del computer MSX.

Settembre 1986 - L. 9.000 - Speciale tutto C64. Inizia da questo numero un'importante serie di programmi didattici sulla matematica nella scuola secondaria superiore. In questa puntata la geometria analitica. Casa automatizzata con C64. I programmi che si acquistano partono in genere in autostart; e i vostri? Quando ci si mette ai comandi di un'astronave di videogame si sa in partenza che il viaggio non sarà affatto tranquillo. Test ministeriali: come prepararsi ad affrontarli vittoriosamente. Il mondo glorioso dei cavalieri di Re Artù in un coloratissimo e originale gioco inglese. Con Memory tieni sotto controllo la gestione del tuo conto in banca e degli interessi maturati. Il comando Dump è tra i più importanti: ecco una routine che ne ottimizza l'uso. Quinta puntata del corso di programmazione in linguaggio macchina: come sfruttare le routine del suo sistema operativo, il kernal.

Ottobre 1986 - L. 9.000 - Speciale tutto C64. Dieta automatica per il lungo inverno, personalizzata e subito pronta da usare. Seconda parte del programma di matematica: è di scena la trigonometria, per verificare calcoli già fatti o per trovare nuove soluzioni. Utility: una routine per formattare più in fretta tabulati e tabelle e una che migliora lo screen editor consentendo di utilizzare il tasto F7 per uscire dal quote mode. Recensioni: un accessorio made in Germany per gestire le eprom e un gioco grandioso, Silent Service. Secondo test per vincere i concorsi pubblici. Gioco: i castelli di cristallo, sfida mozzafiato con eccellente grafica tridimensionale e molti livelli di difficoltà. L'interprete Basic è il protagonista del corso di linguaggio macchina di questo nume.

Novembre 1986 - L. 9.000 - Speciale tutto C64. Terza parte del programma di matematica: sfrutta raffinate nozioni di matematica superiore per lo studio delle funzioni. Utility: MacSimulator per avere un cursore sempre presente sullo schermo e Screen Key per migliorare la gestione del video in Lo-Res. Una eprom e uno switch per migliorare la stampante Mannesmann 802. Il brivido del Casinò a casa vostra con questo gioco che simula un vero tavolo da roulette. Con Superbowl Sunday vestirete i panni di un allenatore di football americano. Inizia da questo numero un corso di inglese adatto a chi ha già un po' di pratica della lingua, con una splendida grafica. Vi presentiamo un progetto straordinario per realizzare un apparecchio che simula una centrale telefonica. Terzo test per i concorsi pubblici dedicato ai candidati laureati. Gioco: un arcade mozzafiato tra bombe, laser e trabocchetti. L'ultima puntata del corso in lm è dedicata alla gestione dell'interrupt. Fast Basic è un'utility in lm che facilita e rende più veloce la stesura di un programma Basic.

Dicembre 1986 - L. 9.000 - Speciale tutto C64. Per gestire al meglio la biblioteca software, ecco un eccezionale archivio elettronico che stampa anche etichette diversificate a seconda del tipo di programma contenuto nella cassetta. Ecco il software che fa del telefonatore del numero scorso una vera centralina telefonica. Le fortezze dello spazio è l'arcade mozzafiato di questo mese, mentre chi vuole un gioco nuovo può farsi da sé un apparecchio speciale con le nostre istruzioni: è un rivelatore di contatti elettrici da collegare al computer, dopo di che... occhio ai contatti!

Gennaio 1987 - L. 9.000 - Tutto per C64. Per velocizzare il drive ecco SpeedDOS. Uno spreadsheet potente e agile in versione su cassetta. Una routine che aggiunge 16 nuovi comandi al Basic, Extended Sprite 3.0. Un classico arcade a stanze, avvincente, in cui un omino deve raccogliere monete tra grotte, sabbie mobili e mille trabocchetti.Ecco Esamina 1.0, un valido aiuto nella ricerca delle poke per la visualizzazione di schermate in Hi-Res. Una missione dura ma non impossibile a bordo di un elicottero da fantascienza; una navicella spaziale sola contro tutti: due giochi avvincenti. Othello, gioco da scacchiera che richiede un'attenta strategia per imprigionare le pedine dell'avversario. Sintesi vocale: hardware e software per dare voce al computer. Un database per catalogare nastri e dischi. Seconda puntata del corso di grafica.

Febbraio 1987 - L. 9.000 - Scrivere con un word processor come questo significa risparmiare un'infinità di tempo. Terza puntata del corso di grafica: sono di scena i set di caratteri. Un test per scoprire la vostra donna ideale. Un'utility che arricchisce la vostra scrivania di eccellenti opzioni grafiche, consentendo di disegnare in scrolling. Database per gestire l'archivio fotografico. Terza puntata del corso di inglese. Due giochi: Sfida nel Pacifico, durante la seconda guerra mondiale e Go for the Gold per cimentarvi in specialità atletiche da Olimpiadi. Grafpad: una tavoletta grafica per C64 facile da usare. Gestione ottimale del registratore con questo hardware che permette il monitoraggio audio e video. Magic Window, otto finestre a scomparsa da poter aprire contemporaneamente sul vostro video.

Marzo 1987 - L. 9.000 - Un database per operazioni veloci e semplici con capienza di mille record. Plotter toolkit: implementa dei comandi per risolvere il problema della compatibilità tra programmi e plotter. Inizia un programma didattico sull'anatomia del corpo umano. Un gioco-quiz sugli animali, per insegnare al computer a riconoscerli. Voice Master: un'interfaccia sonora che consente sintesi di suoni e riconoscimento di parole. Allenatevi al gioco degli scacchi. Una strana Olimpiade che vi vede partecipare a otto originali giochi, caratteristici di otto nazioni. Se amate i videogame alla Rambo, Druid è il gioco che fa per voi, in cui abbondano elementi avventurosi e terrificanti. Disegnatore automatico, per chi disegna schemi elettrici e circuitali, con una ricca biblioteca di simboli. DataMaker, utility in linguaggio macchina che automatizza la stesura delle linee data. Corso di grafica: come si definiscono i caratteri standard, ridefiniti e multicolor.

Tre nella Ram

Si chiama Multi Ram e permette di avere più programmi Basic contemporaneamente in memoria, ognuno indipendente dagli altri e con le proprie risorse individuali.

La caratteristica più stupefacente del Commodore 64 è senza dubbio la sua grande versatilità e malleabilità.

I programmatori più esperti che hanno avuto modo di esplorare a fondo le caratteristiche di questa macchina se ne saranno già resi conto: zone di Ram al riparo da invasioni Basic, possibilità di ridefinire integralmente o in parte l'interprete Basic e completa padronanza della configurazione della memoria.

Multi Ram è un'utility che con un minimo sacrificio di memoria consente di sfruttare una di queste doti del Commodore, la presenza di una zona di Ram inutilizzabile dal Basic, aprendo nuove prospettive di programmazione. Chi ha già provato a programmare in linguaggio macchina avrà sicuramente sfruttato per le proprie routine i 4K di memoria che partono dalla locazione 49152.

La scelta di quest'area viene dettata da due motivi fondamentali. Prima di tutto non richiede nessuna operazione particolare per essere sfruttata e, in secondo luogo, è completamente al riparo dal Basic. Per lungo tempo, quindi, questa zona di memoria è stata considerata il paradiso del programmatore in linguaggio macchina.

Per chi, invece, con il linguaggio macchina non ha mai avuto nulla a che fare quell'area di memoria è sempre stata considerata come un ingiustificato spreco. Multi ram consente di utilizzare da Basic quella zona di Ram, altrimenti invisibile, permettendo inoltre di sezionare in più parti, completamente indipendenti, l'area totale risultante.

Listato

```
$cf08 lda $028f
$cf0b ldx $0290
$cf0e cpx #$cf
$cf10 beq $cf22
$cf12 sta $cff0
$cf15 stx $cff1
$cf18 lda #$23
$cf1a ldx #$cf
$cf1c sta $028f
$cf1f stx $0290
$cf22 rts
$cf23 lda $028d
$cf26 cmp #$04
$cf28 bne $cf3b
$cf2a lda $cb
$cf2c cmp $cfc5
$cf2f beq $cf3b
$cf31 sta $cfc5
$cf34 cmp #$01
$cf36 bne $cf3e
$cf38 jmp $cf9c
$cf3b jmp $(cff0)
$cf3e cmp #$03
$cf40 bcc $cf3b
$cf42 cmp #$07
$cf44 bcs $cf3b
$cf46 ldy $cfc7
$cf49 sta $cfc7
$cf4c lda $2d
$cf4e cpy #$05
$cf50 bcs $cf63
$cf52 ldx $2e
$cf54 sta $cfcc
$cf57 sta $cfcd
$cf5a stx $cfd0
$cf5d stx $cfd1
$cf60 jmp $cf6b
$cf63 sta $cfc9,y
$cf66 lda $2e
$cf68 sta $cfcd,y
$cf6b ldy $cfc5
$cf6e lda $cfc5,y
$cf71 sta $2c
$cf73 lda $cfc9,y
$cf76 sta $2d
$cf78 sta $2f
$cf7a sta $31
$cf7c lda $cfcd,y
$cf7f sta $2e
$cf81 sta $30
$cf83 sta $32
$cf85 lda $cfd1,y
$cf88 sta $38
$cf8a lda #$00
$cf8c tax
$cf8d sta $2b
$cf8f sta ($2b,x)
$cf91 sta $37
$cf93 inx
$cf94 stx $2b
$cf96 lda $cfc5
$cf99 sta $cfc6
$cf9c ldx #$13
$cf9e ldy #$00
$cfa0 lda $cfc5,x
$cfa3 beq $cfad
$cfa5 jsr $cfbb
$cfa8 inx
$cfa9 iny
$cfaa jmp $cfa0
$cfad dey
$cfae dey
$cfaf lda $cfc6
$cfb2 clc
$cfb3 adc #$ad
$cfb5 jsr $cfbb
$cfb8 jmp $(cff0)
$cfbb sta $0400,y
$cfbe lda $0286
$cfc1 sta $d800,y
$cfc4 rts
$cfc5 ---
$cfc6 ---
$cfc7 ---
$cfc8 php
$cfc9 php
$cfca ---
$cfcb cpy #$03
$cfcd ---
$cfce ---
$cfcf ---
$cfd0 php
$cfd1 php
$cfd2 ---
$cfd3 cpy #$a0
$cfd5 ---
$cfd6 ldy #$cf
$cfd8 ldy #$92
$cfda sta ($8d,x)
$cfdc ldy #$bd
$cfde ldy #$a0
$cfe0 ldy #$c7
```

Come funziona

Una volta caricato il programma con Load "Multi Ram" basta dare il consueto RUN. Il programma è costituito da un caricatore Basic che, dopo aver posto i codici della routine in memoria, provvede ad attivarla. A questo punto Multi Ram è pronto per essere utilizzato.

Premendo il tasto CTRL e contemporaneamente il tasto di funzione F1 verrà visualizzata in alto a sinistra sullo schermo la scritta RAM=1. Questo significa che vi trovate nella prima sezione di Ram. Questa sezione è costituita dai primi 8K di memoria dell'area normalmente disponibili al Basic.

Ciò significa che un eventuale programma Basic che verrà editato o caricato sfrutterà solo questi 8K di memoria. Premendo contemporaneamente il

tasto CTRL e il tasto di funzione F3 si entra nella seconda possibile configurazione. In alto sullo schermo verrà visualizzata la scritta RAM=2. In questa situazione ogni programma Basic editato o caricato sfrutterà la seconda metà dell'area normalmente predisposta al Basic.

Premendo CTRL e il tasto di funzione F5 si entrerà nella terza configurazione e in alto sullo schermo verrà visualizzata la scritta RAM=3. A questo punto l'area Basic utilizzabile è costituita dai 3837 byte che seguono la locazione 49152. Come avrete notato lo spazio disponibile in quest'ultima partizione è minore rispetto alle altre, inoltre è inferiore, anche se di poco, ai famosi 4K di Ram liberi (4095 byte per la precisione). I byte che mancano all'appello sono utilizzati dalla routine Multi Ram, che come potete vedere dal **listato 1**, si colloca da $CF 08 a $CFE0 (decimale 53000, 53216) per un totale di soli 216 byte. Per tornare alla configurazione di memoria standard è sufficiente premere CTRL e il tasto di funzione F7. Sullo schermo verrà visualizzata la scritta RAM=0 e l'area Basic utilizzabile sarà costituita dai classici 38911 byte.

Tre programmi

Selezionando una qualsiasi delle configurazioni indicate è possibile creare dei programmi Basic totalmente indipendenti. In tal modo è possibile porre a partire dalla locazione 49152 una serie di routine per leggere la directory e per la creazione di schermate in bassa risoluzione mentre sulle altre due sezioni si possono mettere altri programmi da eseguire o da sviluppare. In ogni istante si può selezionare quale fra questi tre programmi mandare in esecuzione. In nessun caso un programma che risiede in uno dei tre blocchi andrà a influenzare gli altri due. Inoltre le istruzioni Basic NEW e FRE agiranno esclusivamente sul blocco selezionato. Per sapere in quale blocco vi trovate basta premere contemporaneamente i tasti CTRL e RETURN. In alto sullo schermo verrà visualizzata quale area di Ram è attiva in quel momento. Multi Ram accede costantemente a una serie di puntatori aggiornandoli continuamente. Quindi evitate di caricare il programma Multi Ram e di mandarlo in esecuzione quando avete altri programmi in memoria: verrebbero cancellati. Per disattivare Multi Ram basta premere contemporaneamente i tasti RUN/ STOP e RESTORE mentre per riattivarlo basta digitare: SYS 53000.

Silvia Alessi,AndreaRebosio

Sprite editor

Un potente aiuto per la creazione degli sprite: lo Sprite editor. E, inoltre, inizia il viaggio all'interno dell'alta risoluzione.

In questa puntata viene presentato un programma in grado di automatizzare tutte le operazioni necessarie per la definizione degli sprite monocromatici.

Sprite editor è registrato sulla cassetta acclusa alla rivista e per caricarlo basta procedere come al solito digitando load"sprite editor" seguito da Return. La struttura dello Sprite editor proposto è del tutto simile a quella del generatore di caratteri pubblicato nella scorsa puntata; infatti ne condivide tutti i comandi di editing e ha in più la possibilità di ottenere la lista dei codici corrispondenti alla figura che si sta creando.

Per avere la lista completa dei comandi disponibili con i relativi tasti che li attivano basta premere il tasto B. Analizziamo ora in dettaglio tutti i comandi:

• I tasti di movimento del cursore consentono di muoversi all'interno della griglia di definizione dello sprite. Le operazioni di scrittura e cancellazione sono agevolate dalla possibilità di controllare il movimento del cursore sulla griglia anche con i tasti 2, Q, A, e W che permettono di muoversi rispettivamente in alto, a sinistra, in basso e a destra.

• I tasti di funzione F3 e F1 vengono utilizzati rispettivamente per disegnare e cancellare. In ogni istante e automaticamente i due sprite posti alla destra della griglia mostrano il risultato del lavoro che si sta svolgendo. Il primo dei due sprite, quello posto più in alto, è di dimensioni normali mentre il secondo è espanso nelle due direzioni orizzontale e verticale. E' possibile controllare a piacimento le dimensioni di questo secondo sprite premendo il tasto G.

• Premendo il tasto F si attiva il comando di modifica sia del colore di sfondo dello schermo sia del colore in cui viene visualizzato lo sprite. Una volta attivato il comando viene visualizzata la tavolozza di colori con i relativi codici. Prima di inserire il codice numerico corrispondente al colore scelto si deve scegliere se agire sullo sprite o sullo sfondo. Per agire sullo sprite premete il tasto di funzione F3 mentre per agire sullo sfondo premete il tasto F1. Una volta inserito il codice numerico si deve premere il tasto Return.

• I rimanenti comandi di editing sono attivabili con i tasti I, L e V che rispettivamente permettono di invertire, cancellare e far scorrere lo sprite. Il comando di cancellamento deve essere usato con molta cautela poiché non richiede alcuna conferma una volta attivato. Il comando di inversione dello sprite permette di cambiare lo sprite con il suo reverse. Il comando di scorrimento è identico a quello presente sul Character editor della scorsa puntata. Una volta attivato il comando si deve selezionare la direzione verso cui si intende far scorrere lo sprite. La selezione si effettua premendo il tasto posto immediatamente alla destra della direzione desiderata. Una volta effettuato lo scorrimento si esce automaticamente dal comando.

• Load sprite e Save sprite vengono utilizzati per caricare un lavoro già fatto oppure per salvare su disco la definizione di uno sprite e sono attivabili rispettivamente premendo i tasti G e S insieme al tasto CTRL. In entrambi i casi si deve inserire il nome del file e quindi premere il tasto Return.

• Gli ultimi comandi disponibili sono: Mostra directory, Mostra dati e Quit. Il primo si attiva con il tasto C, il secondo con il tasto D e il terzo con il tasto X, unitamente al tasto CTRL. Da notare che la visualizzazione della directory non altera ciò che si trova in memoria e quindi può essere attivato senza particolari accorgimenti. Dopo aver visto la directory si può tornare a disegnare premendo un tasto qualsiasi. Il comando Mostra dati permette di avere la lista completa dei codici corrispondenti allo sprite disegnato. Una volta terminata la lista dei codici si può tornare al modo disegno premendo un tasto qualsiasi. Il comando Quit consente di uscire dal programma e una volta attivato resetta il computer distruggendo Sprite editor. Ovviamente, data la sua pericolosità, richiede una conferma: il tasto S viene interpretato come conferma del comando mentre qualsiasi altro tasto fa ritornare al modo disegno.

L'alta risoluzione

La grafica in alta risoluzione è decisamente più complicata da gestire ri-

Tavola 1. Corrispondenza fra bit e colore

valore dei bit	sorgente del colore
00	locazione53281
01	4 bit più bassi della memoria video
10	4 bit più alti della memoria video
11	memoria del colore

Menù principale dello Sprite Editor.

spetto agli altri modi grafici presentati. Anche visualizzare un solo punto sullo schermo richiederà calcoli piuttosto lunghi e laboriosi.

L'assenza totale di comandi espressamente dedicati alla gestione della grafica si fa sentire in modo particolare proprio quando si ha a che fare con questo modo grafico.

Per tale motivo svilupperemo una serie di routine che faciliteranno la gestione di questa modalità grafica permettendo l'accesso diretto alla pagina grafica.

L'alta risoluzione è un modo grafico come quelli visti nelle precedenti puntate e come tale per essere disponibile deve essere attivato. Sono disponibili due tipi diversi di alta risoluzione, quella monocromatica (in questo caso si parla di standard bit mapped mode) e quella multicolor (detta anche multicolor bit mapped mode).

Il bit 5 del registro 17 del VIC, corrispondénte alla locazione 53265, controlla l'attivazione dell'alta risoluzione monocromatica: se è posto a 1 l'alta risoluzione standard è attivata mentre se è a 0 l'alta risoluzione standard viene disattivata.

Per compiere entrambe le operazioni, attivazione e disattivazione, si può ricorrere alle seguenti istruzioni:

```
POKE 53265, PEEK(53265) OR 32
```

attiva l'alta risoluzione standard

```
POKE 53265, PEEK(53265) AND 223
```

disattiva l'alta risoluzione standard.

Una volta attivata l'alta risoluzione standard si può passare al modo multicolore agendo sul bit 4 del registro 22 del VIC (locazione 53270):

```
POKE 53265, PEEK(53265) OR 32 :
POKE 53270, PEEK(53270) OR 16
```

attiva il modo multicolore

```
POKE 53265, PEEK(53265) AND 223 : POKE 53270, PEEK(53270) AND 239
```

disattiva il modo multicolore.

Come funziona l'Hi-res

Una volta attivata l'alta risoluzione si ha la possibilità di accedere al video utilizzando il punto come più piccola unità indirizzabile. Nella modalità standard lo schermo viene visto come una griglia di 320x200 punti mentre in quella multicolore come una griglia di 160x200; in entrambi i casi c'è la possibilità di accedere individualmente e indipendentemente a ogni punto. Il modo di accesso è del tutto simile a quello che si è utilizzato per la definizione degli sprite o dei caratteri programmabili.

Abbiamo visto che sia con i caratteri programmabili che con gli sprite si ha a disposizione una griglia di definizione che per i caratteri programmabili ha le dimensioni di 8x8 punti mentre per gli sprite di 24x21 punti.

Nel caso dei caratteri programmabili alla griglia corrispondeva una serie di 8 byte consecutivi, uno per ogni riga di otto punti, e il fatto che un punto fosse visibile dipendeva dallo stato del bit corrispondente all'interno del byte.

Lo stesso discorso è rimasto valido per gli sprite con la sola differenza che per ogni riga della matrice di definizione era necessario utilizzare tre byte. Nell'alta risoluzione si ha a che fare con una griglia costituita da 320x200 punti a cui deve corrispondere una codifica in memoria.

Per praticità si può pensare che tale griglia sia costituita da una serie di sottomatrici delle dimensioni di 8x8 punti, cioè di caratteri programmabili (**figura 1**). Ragionando in questi termini la pagina in alta risoluzione risulta suddivisa in 25 righe di 40 caratteri. Infatti i primi 8 byte utilizzati per la codifica della pagina in alta risoluzione sono relativi alle prime 8 righe e non alle prime 8 colonne (**figura 1**).

Le modalità d'accesso ai singoli punti della pagina in alta risoluzione rispecchiano proprio questa struttura ideale. Per rendere visibile un punto dello schermo si deve selezionare il carattere programmabile all'interno del quale si trova il punto, quindi selezionare il byte corrispondente alla linea in cui il punto si trova e, infine, settare il bit corrispondente.

Così come per i caratteri programmabili e per gli sprite anche per l'alta risoluzione si deve disporre di una zona di Ram su cui porre la codifica dell'immagine creata. Dal momento che per coprire il video di caratteri programmabili bisognerebbe disporre di 8K di memoria questo rappresenta proprio il quantitativo di memoria (precisamente di byte consecutivi) necessario per memorizzare la pagina in alta risoluzione.

Esiste tuttavia un ben preciso criterio da seguire nella scelta della zona di memoria: il blocco di 8K deve risiedere all'interno del banco di lavoro del VIC e deve iniziare dalla prima locazione del banco oppure dalla metà del banco.

Ad esempio, nel banco di lavoro di default del VIC (locazioni 0 - 16383) la pagina in alta risoluzione può essere memorizzata a partire dalla locazione 0 oppure a partire dalla locazione 8192 (cioè a 8K dalla prima locazione del banco di lavoro). Per fissare la posizione della codifica della pagina in alta risoluzione si deve agire sul registro 24 del VIC, locazione 53272.

Ecco le istruzioni necessarie per scegliere fra le possibilità a disposizione:

```
POKE 53272, PEEK(53272) OR 8
```

indica che la codifica della pagina in alta risoluzione inizia da metà del ban-

co di lavoro del VIC, cioè a 8K di distanza dalla prima locazione del banco di lavoro (nel caso del banco di lavoro di default la codifica inizierà dalla locazione 8192);

```
POKE 53272, PEEK(53272) AND
247
```

indica che la codifica della pagina in alta risoluzione è memorizzata a partire dalla prima locazione del banco di lavoro del VIC (nel caso del banco di lavoro di default la codifica inizierà dalla locazione 0). E' da notare che, quando l'alta risoluzione viene disattivata, il contenuto del registro 24 deve essere ripristinato con il valore che aveva prima dell'attivazione dell'alta risoluzione.

Il colore nell'alta risoluzione

Il modello dell'alta risoluzione come insieme di caratteri programmabili può essere utilizzato anche per definire le modalità d'accesso al colore. La pagina in alta risoluzione risulta costituita da una serie di caratteri programmabili che nel caso dell'alta risoluzione standard sono caratteri monocromatici mentre in quella multicolore sono caratteri multicolori.

Quindi nell'alta risoluzione standard si hanno a disposizione 2 colori per ogni singolo carattere programmabile ed entrambi indipendenti per ogni carattere; per l'alta risoluzione multicolore si hanno invece a disposizione ben 4 colori per ogni carattere programmabile di cui 3 indipendenti per ogni carattere.

Per controllare il colore nell'alta risoluzione standard si deve agire sulla memoria video (locazioni 1024 - 2023); infatti in questo modo grafico la memoria video si comporta come la memoria del colore nella bassa risoluzione: la prima locazione, cioè la 1024, controlla il colore del carattere programmabile posto al di sotto e così dicasi per tutte le altre locazioni. Per la memorizzazione del colore si deve tenere presente che i 4 bit meno significativi (bit 0 - 3) della memoria video controllano il colore dei punti invisibili del corrispondente carattere programmabile mentre i rimanenti 4 bit (bit 4 - 7) controllano il colore dei punti visibili. Per esempio l'istruzione POKE 1024,3 ha come effetto, una volta attivata l'alta risoluzione standard, di fissare i colori ciano e nero rispettivamene per i punti invisibili e visibili del carattere programmabile posto in alto a sinistra, cioè nella posizione che, nel modo testo, corrisponde alla locazione video 1024.

Una nota molto importante: la corrispondenza fra memoria video e pagina in alta risoluzione è indipendente dalla posizione in memoria di quest'ultima; quindi la locazione 1024 controllerà sempre il colore di visualizzazione dei punti corrispodenti ai primi 8 byte dell'alta risoluzione; la locazione 1025 controllerà il colore dei punti corrispondenti ai successivi 8 byte e così via per tutte le altre locazioni della memoria video.

Esempio di sprite in costruzione.

Nell'alta risoluzione multicolore i colori sono controllati dalla memoria video, dalla memoria del colore (locazioni 55296 - 56295) e dal registro 33 del VIC (locazione 53281).

Ogni coppia di punti è controllata da una corrispondente coppia di bit e lo stato di questi bit indica la sorgente del colore. Nella **tavola 1** viene mostrata la corrispondenza fra la coppia di bit e le sorgenti di colore.

E' da notare che, nell'alta risoluzione multicolore, la suddivisione della pagina grafica in coppie di punti non è arbitraria ma prefissata; in pratica una coppia di punti è accessibile solo se a essa corrisponde una delle seguenti coppie di bit: 0,1 - 2,3 - 4,5 - 6,7. Questo fatto deve essere tenuto in considerazione per evitare risultati inspiegabili.

Un esempio pratico

Supponiamo di voler utilizzare l'alta risoluzione e di impiegare gli 8K di memoria che vanno dalla locazione 8192 fino alla locazione 16383 (il banco di lavoro del VIC è quello di default) per memorizzare la codifica della pagina in alta risoluzione. Con una simile scelta si hanno a disposizione

Riga	Bit	Colonna 0 76543210	Colonna1 76543210	Colonna 2 76543210	ecc.
	byte0	....	....	....	
0	byte1	....	....	....	
	byte2	....	....	....	
	byte3	....	....	....	
	byte4	....	....	....	
	byte5	....	....	....	
	byte6	....	....	....	
	byte7	....	....	....	
Riga					
	byte320	....	....	....	
1	byte321	....	....	....	
	byte322	....	....	....	
	ecc.				

Figura 1. Codifica in memoria della pagina grafica.

circa 6K di memoria per un eventuale programma Basic che sfrutti l'alta risoluzione.

La prima operazione da compiere, una volta attivata l'alta risoluzione, consiste nel pulire l'area di memoria che contiene la codifica della pagina in alta risoluzione e nell'inizializzare la memoria video con i colori desiderati. Una volta effettuate tutte le operazioni indicate non resta che visualizzare qualcosa. Per rendere visibile oppure per cancellare un punto della pagina in alta risoluzione è necessario individuare il bit ad esso corrispondente all'interno della memoria riservata per la codifica.

Ricollegandoci al discorso fatto in precedenza, si deve innanzitutto trovare il carattere programmabile a cui il punto appartiene; quindi individuare la riga all'interno della quale si trova il punto da settare. Per compiere queste operazioni si deve fare riferimento a una formula. Si è detto che la pagina in alta risoluzione è costituita da una matrice di 320 punti orizzontali e 200 verticali (che nell'alta risoluzione multicolore sono visti a coppie). Ciascun punto di questa matrice può essere individuato dalle sue coordinate, una volta che si è fissato un punto di riferimento. Se indichiamo con le coordinate (0,0) l'angolo superiore sinistro dello schermo e con X e Y le coordinate del punto che intendiamo settare, la riga in cui questo punto si trova è:

RIGA = INT(Y/8)

RIGA rappresenta la riga all'interno della quale si trova il carattere programmabile che contiene il punto da settare; infatti se si fanno alcuni rapidi calcoli si scopre che RIGA assume solo un valore compreso fra 0 e 24, appunto i valori che individuano le righe della memoria video. Il carattere corretto si può individuare così:

CARATTERE = INT(X/8)

CARATTERE rappresenta il blocco di 8 byte all'interno del quale si trova quello che contiene il bit corrispondente al punto da settare; infatti CARATTERE può assumere solo un valore compreso fra 0 e 39, appunto i valori che corrispondono alle colonne della memoria video. Una volta individuato il carattere programmabile si deve selezionare la riga della matrice di definizione in cui si trova il punto:

LINEA = Y AND 7

LINEA rappresenta la linea della matrice di definizione del carattere programmabile all'interno della quale si trova il punto da settare. Ora non resta che trovare il punto all'interno della linea:

BIT = 7 - (X AND 7)

BIT rappresenta il bit corrispondente al punto da visualizzare e può assumere solo un valore compreso fra 0 e 7, appunto i valori che individuano un bit all'interno di un byte. Mettiamo insieme tutte le singole operazioni in una sola formula che ci permet-

Listato 1.

```
  100 rem -----
  110 rem
  120 rem hires
  130 rem
  140 rem -----
  150 :
  160 v=53248:rem indirizzo base del vic
  170 sa=8192:rem indirizzo inizio pagina in alta risoluzione
  180 pokev+33,11:pokev+32,11: rem set colore bordo e sfondo
  190 gosub 900: rem attiva l'alta risoluzione
  200 gosub 1060: rem pulisce la pagina in alta risoluzione
  210 co=7*16+11:gosub 1100:rem setta i colori per l'alta risoluzione
monocromatica
  220 :
  230 rem -- punti --
  240 :
  250 b0=1:b1=0:forxc=0 to 100: rem disegna il grafico del seno
  260 yc=50*sin(xc/25.5)+99
  270 gosub 1300:next
  280 :
  290 rem -- linea --
  300 :
  310 x1=90:y1=120:x2=110:y2=50
  320 gosub 1460
  330 :
  340 rem -- cerchio --
  350 :
  360 xr=40:yr=40:xm=160:ym=100
  370 gosub 1620
  380 :
  390 rem -- ellisse --
  400 :
  410 xr=40:yr=20:xm=260:ym=100
  420 gosub 1620
  430 wait198,255
  440 :
  450 rem ----------
  460 rem multicolor
  470 rem ----------
  480 :
  490 rem ------------------
  500 rem fissa altezza barre
  510 rem ------------------
  520 :
  530 m=1:l(0)=20:l(1)=30:l(2)=5:l(3)=26:l(4)=23:l(5)=10
  540 :
  550 rem ----------------------
  560 rem pulisce la pagina hires
  570 rem ----------------------
  580 :
  590 gosub 1070
  600 :
  610 rem ------------------------------
  620 rem attiva l'alta ris. multicolore
  630 rem ------------------------------
  640 :
  650 pokev+22,peek(v+22)or16
  660 :
  670 rem ------------------------------
  680 rem fissa i colori per multicolore
  690 rem ------------------------------
  700 :
  710 co=6*16+3:cp=0:gosub1100
  720 :
  730 rem --------------
  740 rem visualizza assi
  750 rem --------------
  760 :
  770 b0=1:b1=1:yc=100:for xc=0 to 318 step2:gosub1390:next
  780 xc=20:for yc=0 to 200:gosub1390:next:jj=40
  790 :
  800 rem ---------------
  810 rem visualizza barre
  820 rem ---------------
  830 :
  840 yc=101:for is=0 to 5:fork=0 to l(is)
  850 xc=jj:b0=0:b1=1:yc=yc-1:fort=1 to 3 :gosub 1390:xc=xc+2:next:xc=jj
  860 b0=1:b1=0:yc=yc-1:fort=1 to 3:gosub 1390:xc=xc+2:next
  870 nextk:yc=101:jj=jj+20:next
  880 poke198,0:wait198,255:gosub1260:print"[clr]":end
  890 :
```

te, una volta assegnate le coordinate di individuare il byte su cui si deve agire per rendere visibile il punto:

BYTE = BASE + RIGA * 320 + CARATTERE * 8 + LINEA

dove BASE rappresenta l'indirizzo di inizio della zona destinata a contenere la codifica della pagina. Si setta con:

POKE BYTE, PEEK(BYTE) OR 2^BIT

Le istruzioni appena viste vanno bene solo per l'alta risoluzione monocromatica.

Per rendere la formula utilizzabile anche per l'alta risoluzione multicolore si deve modificare l'istruzione che calcola BIT:

BIT = (7-(X AND 7))*B0+(7-((X+1) AND 7))*B1

dove B0 e B1 rappresentano lo stato dei bit che corrispondono alla coppia di punti da visualizzare. Ad esempio se si vuole visualizzare una coppia di punti utilizzando come sorgente del colore la memoria del colore B0 e B1 devono valere 1. Nel caso in cui si stia utilizzando l'alta risoluzione monocromatica B0 deve sempre valere 1 mentre B1 deve valere 0.

Il **listato 1** contiene un programma dimostrativo per l'alta risoluzione standard e multicolore. Dato il RUN il programma visualizza, in alta risoluzione standard, parte del grafico della funzione seno, una retta, un cerchio e un'ellisse.

Una volta terminata la visualizzazione delle figure, si deve premere un tasto per vedere il seguito.

A questo punto viene attivata l'alta risoluzione multicolore (bisogna aspettare alcuni secondi perché la pagina in alta risoluzione deve essere nuovamente pulita) e vengono visualizzati dei grafici a barre.

Terminata la visualizzazione delle barre (sono in tutto 6) basta premere un tasto per tornare al modo testo.

Le routine che provvedono al calcolo delle cordinate per la visualizzazione della retta e dell'ellisse sono strutturate come sottoprogrammi e come tali possono essere utilizzate per programmi di uso più generale.

A tal proposito le analizzeremo brevemente:

• Linee 1460 - 1600. Contengono la routine per la visualizzazione di una linea. Senza dilungarci troppo sull'algoritmo utilizzato è sufficiente dire che la routine traccia una retta fra i punti di coordinate (X1,Y1) e (X2,Y2)

• Linee 1620 - 1720. Questa è la routine per la visualizzazione di una ellisse. E' la stessa routine utilizzata per la visualizzazione di un cerchio, che è un caso particolare di ellisse.

I parametri da passare alla routine sono: XR,YR,XM,YM che, rispettivamente, rappresentano i semiassi maggiore e minore dell'ellisse (XR= YR nel caso della circonferenza) e le coordinate del centro dell'ellisse.

Paolo Gussoni
(continua)

```
 900 rem -------------------------
 910 rem
 920 rem attiva l'alta risoluzione
 930 rem
 940 rem -------------------------
 950 :
 960 poke v+24,peek(v+24) or 8: rem inizio pagina in alta risoluzione
     a 8192
 970 poke v+17,peek(v+17) or 32: rem attiva l'alta risoluzione
 980 poke v+22,peek(v+22) and 239: rem disattiva il modo multicolore
 990 return
1000 :
1010 rem ------------------------------
1020 rem
1030 rem pulisce la pagina in alta ris.
1040 rem
1050 rem ------------------------------
1060 :
1070 forx=sa to sa+7999:pokex,0:next
1080 return
1090 :
1100 rem ------------------------------
1110 rem
1120 rem setta i colori per l'alta ris.
1130 rem
1140 rem ------------------------------
1150 :
1160 bc=1024:forx=bc to bc+999:pokex,co:next
1170 if m=1 then mc=55296:forx=mc to mc+999:pokex,cp:next
1180 return
1190 :
1200 rem ---------------------------
1210 rem
1220 rem disattiva l'alta risoluzione
1230 rem
1240 rem ---------------------------
1250 :
1260 pokev+17,peek(v+17) and 223: rem disattiva l'alta risoluzione
1270 pokev+22,peek(v+22) and 239: rem disattiva il modo multicolore
1280 poke v+24,peek(v+24) and 255-8: rem set di caratteri in 4096
1290 return
1300 :
1310 rem -------------------------
1320 rem
1330 rem calcola le coordinate
1340 rem
1350 rem del punto da visualizzare
1360 rem
1370 rem -------------------------
1380 :
1390 ra=320*int(yc/8)+(yc and 7)
1400 ba=8*int(xc/8)
1410 ma=(2^(7-(xc and 7)))*b0+2^(7-((xc+1) and 7))*b1
1420 ad=sa+ra+ba
1430 poke ad,peek(ad) or ma
1440 return
1450 :
1460 rem -----------------
1470 rem
1480 rem disegna una linea
1490 rem
1500 rem -----------------
1510 :
1520 dy=y2-y1:dx=x2-x1
1530 yc=y2:xc=x2
1540 if dx=0 then foryc=y2 to y1 step sgn(-dy):gosub1390:next:goto1600
1550 dd=dy/dx
1560 for xc=x2 to x1 step sgn(-dx)
1570 tc=int(dd*(xc-x2)+y2)
1580 if tc<>yc then yc=yc+sgn(-dy):gosub1390:goto 1580
1590 gosub 1390:next xc
1600 return
1610 :
1620 rem ---------------
1630 rem
1640 rem disegna ellisse
1650 rem
1660 rem ---------------
1670 :
1680 forf2=-1 to 1 step 2:rem flag sinistra/destra
1690 for x=0 to f2*xr step f2
1700 tc=yr*sqr(1-(x/xr)^2):xc=x+xm: rem calcola il punto
1710 yc=ym+tc:gosub 1390:yc=ym-tc:gosub 1390
1720 nextx,f2:return
```

Mille luci

Ecco Commolight, il programma per gestire al meglio l'omonimo apparecchio presentato sullo scorso numero. Dispone di quattro canali autonomi.

Nello scorso numero *Radio Elettronica & Computer* ha presentato ai lettori Commolight, un simpatico hardware per la gestione di un parco lampade. Il programma che trovate nella cassetta allegata a questo numero, usato appunto con l'apparecchio Commolight, è in grado di trasformare il Commodore 64 in una vera e propria centrale elettronica capace di pilotare un parco lampade di 4 canali distinti collegato alle uscite del Commolight.

Sono previsti due modi di funzionamento: Auto, che abilita il clock interno, e Music per l'elaborazione degli effetti al ritmo del segnale audio eventualmente collegato in input.

Il programma si carica sul Commodore 64 con i consueti comandi di Loading, ovvero LOAD"" oppure LOAD COMMOLIGHT o anche con il più pratico metodo di pressione contemporanea dei tasti C= (logo Commodore) e RUN STOP.

Come funziona

Al centro dello schermo è disegnato il riquadro del display con le quattro uscite A, B, C e D e i relativi segnalatori colorati, indicativi di quello che dovrebbe essere il colore delle lampade collegate agli stadi finali, al fine di ottimizzare le prestazioni: in particolare sono consigliabili tonalità bianche o gialle per le uscite A e D (frequenze audio basse), mentre per le uscite centrali B e C (medie frequenze audio), è preferibile usare verde, rosso o blu.

Nella parte inferiore del video si trovano i 3 menù delle opzioni:

• **Il primo menù**, azzurro, permette di scegliere il tipo di clock per la modulazione:

F1. Auto
F3. Music.

Premendo il tasto F1 si seleziona il clock interno automatico (funzione Auto) che permette l'esecuzione degli effetti anche senza segnali audio; con F3 si abilita invece il funzionamento a ritmo musicale (Music) e gli effetti visibili dipendono in gran parte dal segnale musicale presente in input sul Commolight, e dal relativo volume (potenziometro Level): ovviamente Music si seleziona solo quando c'è la reale presenza di segnale audio (anche da microfono collegato alla presa Micro) altrimenti, non essendoci attivazione del clock, le sequenze non vengono riprodotte.

• **Il secondo menù**, di colore verde, ha efficacia solo se è attivata la funzione Auto.

Permette di scegliere la frequenza di modulazione, normale o velocizzata:

F5 . Clock+
F7. Clock -

Premendo F5 si seleziona il modo veloce (Clock+) e gli effetti vengono riprodotti molto velocemente; con F7 si opta invece per la frequenza normale (Clock-).

• **Il terzo menù**, di colore grigio, è il più ampio in quanto contiene ben 7 opzioni distinte che sono in pratica gli effetti esecutivi da selezionare:

1. Psico
2. Strobe
3. Shift
4. Flash
5. Total
6. Single
7. Light jockey.

1. **Psico**. E' il tradizionale ed efficace modo di visualizzazione psichedelico: in particolare le uscite A e D corrispondono alle basse frequenze audio mentre C e D rappresentano le medie tonalità.

2. **Strobe**. E' una sequenza up-down bidirezionale molto originale.

3. **Shift**. E' una completa sequenza preprogrammata comprendente svariatissimi giochi di luce che sfruttano al massimo i 4 canali disponibili.

4. **Flash**. Lampeggio contemporaneo dei canali ad altissima frequenza.

5. **Total**. Vu-meter proporzionale a 4 livelli: aumenta e decresce in base al segnale presente in input (se è attivata la funzione Music).

6. **Single**. E' un effetto molto suggestivo di luci scorrevoli da sinistra verso destra (guardando il video) ad alta frequenza.

7. **Light Jockey.** E' di grande utilità ed efficacia in quanto permette di esplorare in sequenza tutti i 6 suddetti effetti senza bisogno di rimanere alla tastiera a premere pulsanti effettuando continuamente scelte e variazioni: funziona sia in Music sia con Auto + Clock+ sia con Auto+ Clock-.

Con tutti i 6 effetti si ottiene un ottimo output selezionando la funzione Music, particolarmente efficaci con detta modulazione a ritmo di musica sono Psico, Flash e Total; con la funzione Auto abbinata a Clock è possibile invece apprezzare Shift e Total mentre con Auto e Clock sono ottimizzati Psico e Single.

Daniele Malavasi

Vorrei sapere, vorrei proporre...

Quick Base non allinea

Ho constatato che il programma di archiviazione Quick Base pubblicato sul numero di marzo, pur mantenendo in memoria correttamente sia campi numerici sia campi testo, dopo il salvataggio su disco e la successiva rilettura del file allinea i campi numerici a sinistra come i testi, e di conseguenza trascina tutti i caratteri dei campi successivi. E' sufficiente qualche modifica per ovviare a questo inconveniente?

Gianni Raschini
Torino

L'errore è dovuto all'istruzione INPUT utilizzata per leggere i dati da disco. Come è noto se una stringa ha degli spazi in testa, come i campi numerici allineati a destra, l'istruzione INPUT li ignora, alterando così tutti campi successivi. Il modo più semplice di evitare tutto ciò è di inserire la stringa da salvare su disco tra doppi apici (le virgolette, codice ASCII 34) in modo tale che eventuali spazi in testa vengano inglobati nella stringa letta. Quindi basta modificare la linea 420 in questo modo:

420 PRINT#1,Y:FORI=1TOY: PRINT#1,CHR$(34)A$(I)CHR$(34) :NEXTI

Basic senza timer

Seguo assiduamente la vostra rivista perché trovo sempre utili suggerimenti per tutti i programmatori che come me hanno imparato (e stanno imparando) strada facendo tutti i trucchi per utilizzare al meglio il C64. Per il momento programmo solo in Basic e quindi non sempre riesco a ottenere i risultati che mi sono proposto perché in alcuni casi le routine in linguaggio macchina sono davvero indispensabili. Per esempio un problema molto semplice, ma irrisolvibile in Basic, è quello dell'orologio. Molti programmi necessitano infatti di un cronometro sempre attivo sullo schermo, ma mantenerlo aggiornato senza rallentare il programma è impossibile. Inoltre i continui richiami alla routine dell'orologio allungano e complicano il programma principale. Avreste una routine in LM ad hoc anche per questo problema?

Roberto Di Leonardo
Novara

Il problema è un classico, infatti per essere risolto adeguatamente necessita di una routine che non solo sia scritta in linguaggio macchina, ma anche lavori da interruzione. Lo scopo che si vuole conseguire implica due cose:

1. L'aggiornamento e la visualizzazione devono essere tanto veloci da non rallentare il programma principale (a questo basta il linguaggio macchina).

2. La routine non deve essere chiamata esplicitamente dal programma, per esempio con una SYS, ma dovrebbe aggiornare l'orologio a intervalli di tempo regolari (per questo è necessario l'uso delle interruzioni).

Il programma di aggiornamento in definitiva è sempre un loop. Il problema è decidere a quale livello implementare il loop. Nel caso del Basic, il ciclo è nel programma principale (aggiornamento a controllo di programma) nel caso della routine da interruzione il ciclo è interno a quello eseguito dal microprogramma che con gestisce la CPU.

Ogni sessantesimo di secondo l'hardware provoca un'interruzione, cioè un particolare segnale che costringe la CPU a salvare il proprio stato (registri base e contatore di programma) e a mandare in esecuzione una routine predefinita, per esempio per la scansione della tastiera. E' sufficiente inserire un richiamo alla routine di aggiornamento dell'orologio a questo livello e il gioco è fatto. Ogni sessantesimo di secondo l'orologio sullo schermo varrà aggiornato, indipendentemente da ciò che accade nel programma Basic.

*Qui sotto riportiamo il programma che realizza questa funzione, sia nella versione caricatore Basic (**listato 1**) che in quella LM (**listato 2**).*

Listato 1. Timer in Basic

```
100 rem --------------------------------
110 rem -                              -
120 rem -                              -
130 rem -      timer da interrupt      -
140 rem -                              -
150 rem -                              -
160 rem --------------------------------
170 poke53280,0:poke53281,0
180 print"[clr][yel][down][down][down][down][down][down][down][down][d
    own][down][down]"spc(15)"attendi..."
190 :
200 deffna(x)=int(x/10)*16+(x-int(x/10)*10)
210 :
220 poke792,193:ad=49152:forc=0to91:readh$:l=asc(left$(h$,1))-48
230 r=asc(right$(h$,1))-48:pokead+c,(l+(l>9)*7)*16+(r+(r>9)*7):next
240 dataa2,00,ad,0b,dc,29,7f,20,00,00,a2,03,ad,0a,dc,20,00,00,a2,06,ad
    ,09,dc
250 data20,00,00,ad,08,dc,a9,ba,8d,02,04,8d,05,04,a9,07,8d,02,d8,8d,05
    ,d8
260 data4c,31,ea,48,29,f0,4a,4a,4a,4a,18,69,b0,9d,00,04,68,29,0f,18,69
    ,b0
270 data9d,01,04,a9,07,9d,00,d8,9d,01,d8,60:rem byte 07=colore orologi
    o
280 data78,a9,00,8d,14,03,a9,00,8d,15,03,58,60
290 su=ad+48:sh=int(su/256):sl=su-sh*256:jh=int(ad/256):jl=ad-jh*256
300 pokead+8,sl:pokead+9,sh:pokead+16,sl:pokead+17,sh:pokead+24,sl:pok
    ead+25,sh
310 pokead+81,jl:pokead+86,jh:poke56334,peek(56334)or128:sysad+79
320 poke56335,peek(56335)and127
330 poke56331,fna(00):rem ore
340 poke56330,fna(00):rem minuti
350 poke56329,fna(00):rem secondi
360 poke56328,0:print"[clr][down][down][down][down][down][down][down][
    down][down]"
```

Listato 2. Timer in LM

```
$c000 ldx #$00
$c002 lda $dc0b
$c005 and #$7f
$c007 jsr $c030
$c00a ldx #$03
$c00c lda $dc0a
$c00f jsr $c030
$c012 ldx #$06
$c014 lda $dc09
$c017 jsr $c030
$c01a lda $dc08
$c01d lda #$ba
$c01f sta $0402
$c022 sta $0405
$c025 lda #$07
$c027 sta $d802
$c02a sta $d805
$c02d jmp $ea31
$c030 pha
$c031 and #$f0
$c033 lsr
$c034 lsr
$c035 lsr
$c036 lsr
$c037 clc
$c038 adc #$b0
$c03a sta $0400,x
$c03d pla
$c03e and #$0f
$c040 clc
$c041 adc #$b0
$c043 sta $0401,x
$c046 lda #$07
$c048 sta $d800,x
$c04b sta $d801,x
$c04e rts
$c04f sei
$c050 lda #$00
$c052 sta $0314
$c055 lda #$c0
$c057 sta $0315
$c05a cli
$c05b rts
```

Vorrei sapere, vorrei proporre...

La routine in lm occupa esattamente 92 byte e risiede nei 4 K dopo l'interprete e quindi non occupa memoria Basic. Viene attivata con SYS49204 e visualizza l'orologio nell'angolo in alto a sinistra. Da notare la funzione a(x) definita alla linea 200 che consente di impostare l'orologio: linee 330-350. Inoltre alla linea 270 il byte 07 indica il colore con cui deve essere visualizzato l'orologio, in questo caso giallo.

Usare le finestre

Da quasi un anno compro volentieri la vostra rivista che trovo molto interessante. Posseggo un C64 completo di disk drive e stampante e nei ritagli di tempo mi piace programmare.

Il problema che desideravo sottoporvi è il seguente: nel numero di febbraio 87 ho trovato il programma Magic Window che ritengo veramente utile, tanto da decidere di adottarlo e usarlo in alcuni miei programmi. Purtroppo non sono riuscito né a trasferire il programma su disco né tantomeno a inserirlo in un mio programma. Potreste suggerirmi come fare?

Nico Ascanio
Treviso

Magic Window è scritto interamente in linguaggio macchina e risiede in memoria da $C000 a $CCC8 (in decimale da 49152 a 52424). Per questo motivo una volta caricato in memoria non può essere salvato come fosse un normaleprogrammaBasic.E' necessario utilizzare un monitor per il linguaggio macchina che dispone del comandonecessarioper salvareunaqualsiasi area di RAM. In particolare lo ZOOM rilocato in 4096 o in 32768 va bene per questo scopo. L'istruzione è la seguente: S "MAGIC WINDOW",08,C000,CCC8.

Una volta creato il file su disco è molto semplice inglobarlo in un programma Basic, tenendo conto che l'espansione viene attivata con SYS 49152. Ecco un esempio:

```
100 if a=0 then a=1: load"magic
window",8,1
110 sys 49152<resto del programma>
```

Naturalmente è possibile fare delle versioni un po' più compatte, costituite da un solo file che contiene sia il programma sia l'espansione, utilizzando una cartuccia con l'opzione di frezee come Isepic, final cartridge.

Sintesi sì, ma come?

Ho ordinato il circuito hardware della sintesi vocale e mi è arrivato poco tempo fa. Purtroppo ho dei problemi. Tutto funziona bene fino a quando decido di utilizzare il master in un mio programma Basic. Per dire la verità non l'ho utilizzato ancora in un vero programma, perchè già provando in modo diretto, appena introduco la POKE 52824,F dopo aver caricato il master (non superiore a 12K) il C64 mi dà il messaggio OUT OF MEMORY ERROR e senza nemmeno aver scritto il numero della parola! Anche inserendo il numero della parola e la SYS il C64 imperterrito mi fornisce sempre lo stesso messaggio di errore.

Qual è il rimedio? Inoltre non ho ben capito come inserire in un programma Basic il file con i messaggi digitalizzati: bisogna caricare il master prima del programma o includere il master nel programma stesso? Come? Infine come funziona l'ordinamento delle frasi e il loro relativo richiamo da Basic?

Renato Ercoli
Roma

Il messaggio di OUT OF MEMORY è causato da un bug nel sistema operativo del C64 che obbliga a dare una NEW ogni qual volta viene caricato un file in linguaggio macchina che si colloca fuori dall'area Basic, come nel caso del file che contiene le frasi digitalizzate. Quindi se si vuole utilizzare in modo diretto il file mster è necessario dopo il caricamento dare una NEW. Se invece si vuole utilizzare il file master direttamente all'interno di un programma, la cosa più semplice è quella di caricare il file esterno come fosse una semplice routine in lm:

```
100 if a=0 then a=1: load"master",8,1
110 rem resto del programma
```

In questo caso si è fatta l'ipotesi di un caricamento da disco.

Appena il programma viene lanciato, viene caricato in memoria il file master (questa volta senza rischio di messaggi di errore perchè in modo programma) e successivamente, grazie alla variabile flag A viene eseguito il resto del programma.

Per costruire un file master, cioè un file contenente tutte le frasi che si vogliono inserire in uno stesso programma, a partire dalle singole registrazioni, è opportuno seguire questi passi:

1. Selezionare opportunamente i parametri sul menu principale (pagine disponibili e frequenza di campionatura).

2. Effettuare tutte le registrazioni e salvarle singolarmente su disco o su nastro con l'opzione 5.

3. Selezionare l'opzione 6 per creare un file master e indicare via via tutti i file delle registrazioni che vanno inseriti. E' opportuno scriversi l'ordine con cui sono state collegate le varie registrazioni perchè questo sarà lo stesso ordine con cui verranno richiamate da Basic. All'interno di un programma Basic, una volta caricato in memoria il file master, per ascoltare una delle registrazioni è sufficiente eseguire le seguenti istruzioni:
POKE52824,R: SYS52824

Dove R è il numero della registrazione all'interno del file master (vedi passo 3). La prima registrazione avrà associato il numero 0. Un esempio di questo modo di procedere è fornito nel demo presente sulla cassetta:

```
100 a=52824:s=a+1
110 geta$:ifa$=""then110
120 ifa$="1"thenb=0:goto220
130 ifa$="2"thenb=1:goto220
140 ifa$="3"thenb=2:goto220
<controlli sugli altri tasti>
220 pokea,b:syss:goto100
```

Alla pressione di un tasto numerico viene richiamata la registrazione che contiene la parola corrispondente. Durante l'emissione della voce il bordo lampeggia e il programma Basic attende che l'emissione termini.

Caricate così i programmi della cassetta allegata a questo numero di *Radio Elettronica&Computer:*

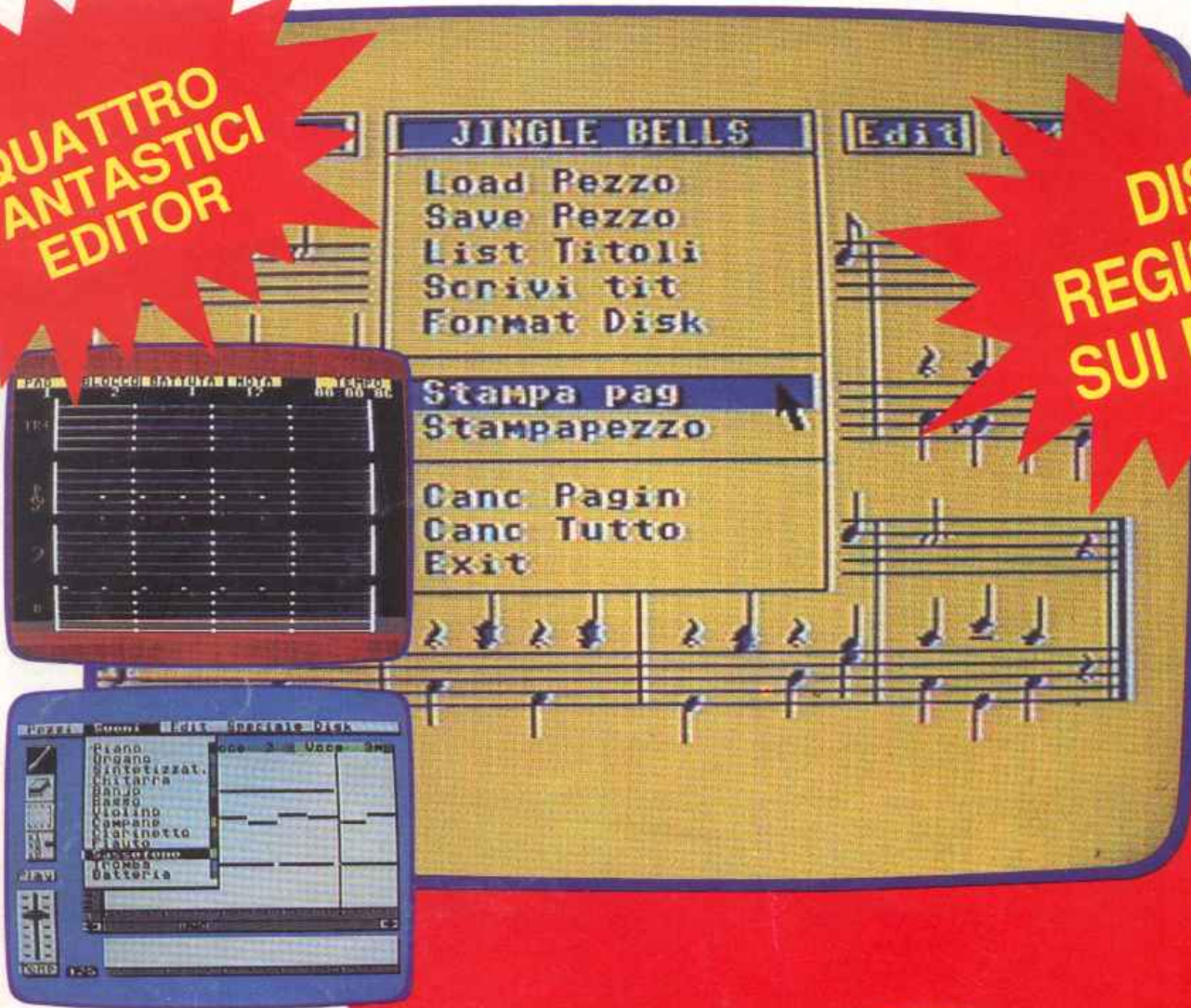
Riavvolgete il nastro e digitate LOAD seguito da Return sulla tastiera del C64 e PLAY sul registratore. Verrà caricato il programma di presentazione con il menù dei programmi. Digitate RUN seguito dalla pressione del tasto RETURN. Terminata la presentazione, per caricare uno qualsiasi dei programmi è sufficiente digitare: LOAD "NOME PROGRAMMA" seguito dalla pressione del tasto RETURN.

Sped. in Abb. Postale Gr. III/70%

Tutto COMMODORE Musica

Anno I - Numero 3 - MAGGIO 1987 - L. 13.000

QUATTRO FANTASTICI EDITOR

DISCO REGISTRATO SUI DUE LATI

TASSA PAGATA PER CAMPIONE ALLEGATO

COMPOSER

MUSIC MASTER

BATTERISTA

SUPER SPARTITO